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*

1.0.1 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 1.0.3) and the different ways in which these can help convey information to the reader (Subsection 1.0.5, Subsection 1.0.8). 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 (Subsection 1.0.11).

1.0.2 Networks And Their Role In Visual Analytics

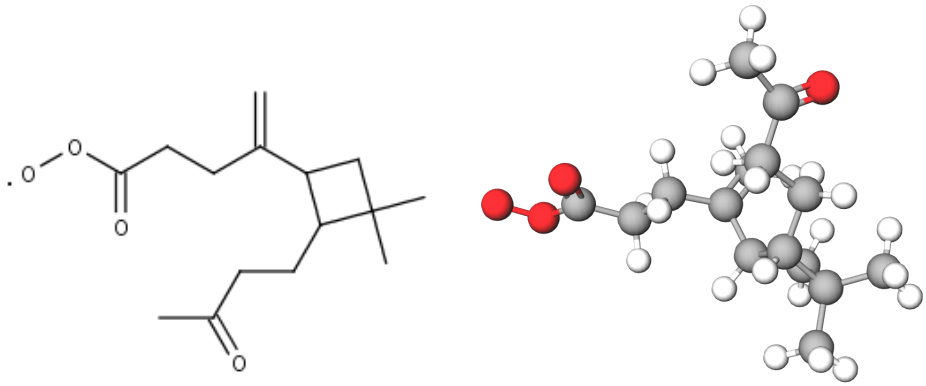
Networks are present everywhere - this ranges from interactions within social media to bank transactions, internet routing, genetics to epidemiology [Martin Grandjean, 2016; Staples et al., 2013; **?**; Baronchelli et al., 2013; Sangers et al., 2019; Kohlbacher et al., 2014; Archambault et al., 2014; 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 edges or links [Kerren et al., 2014]. These terms can be used interchangeably and will do, throughout this thesis.

1.0.3 Graphs In Chemistry

node-link representations have been a core part of the field of chemistry for many years, showing the types of bonds between different atoms. They are integral in the representation of molecules, using the ball-stick (graph) style analogy, both physically (with the aid of molecular model kits) or pictorially to show various structural properties (Figure 1.1). Using such analogies aid in the tasks of identifying the features, functional groups and bond properties of the species and how they can react.

1.0.3.1 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 un-



(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].

dergo. In general, this may be done using a series of interconnected reactions in the form of a reaction cycle (as shown in 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.

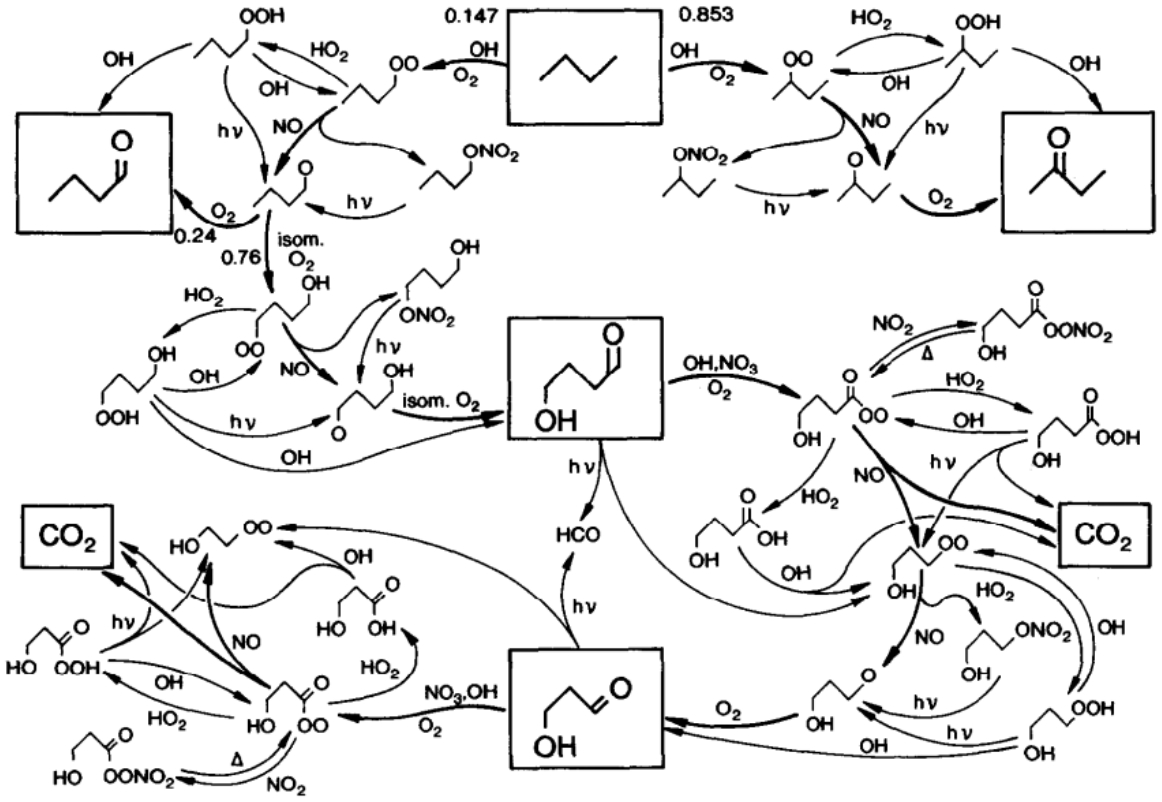


Figure 1.2: **A systematic representation of the degregation of butane.** Using this we are able to see the process C_4H_{10} undergoes before its ultimate demise as carbon monoxide and water. Source: [Jenkin et al., 1997]

1.0.4 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.

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 (Subsection 1.0.5) and semantics (Subsection 1.0.8).

1.0.5 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 1.3. 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.

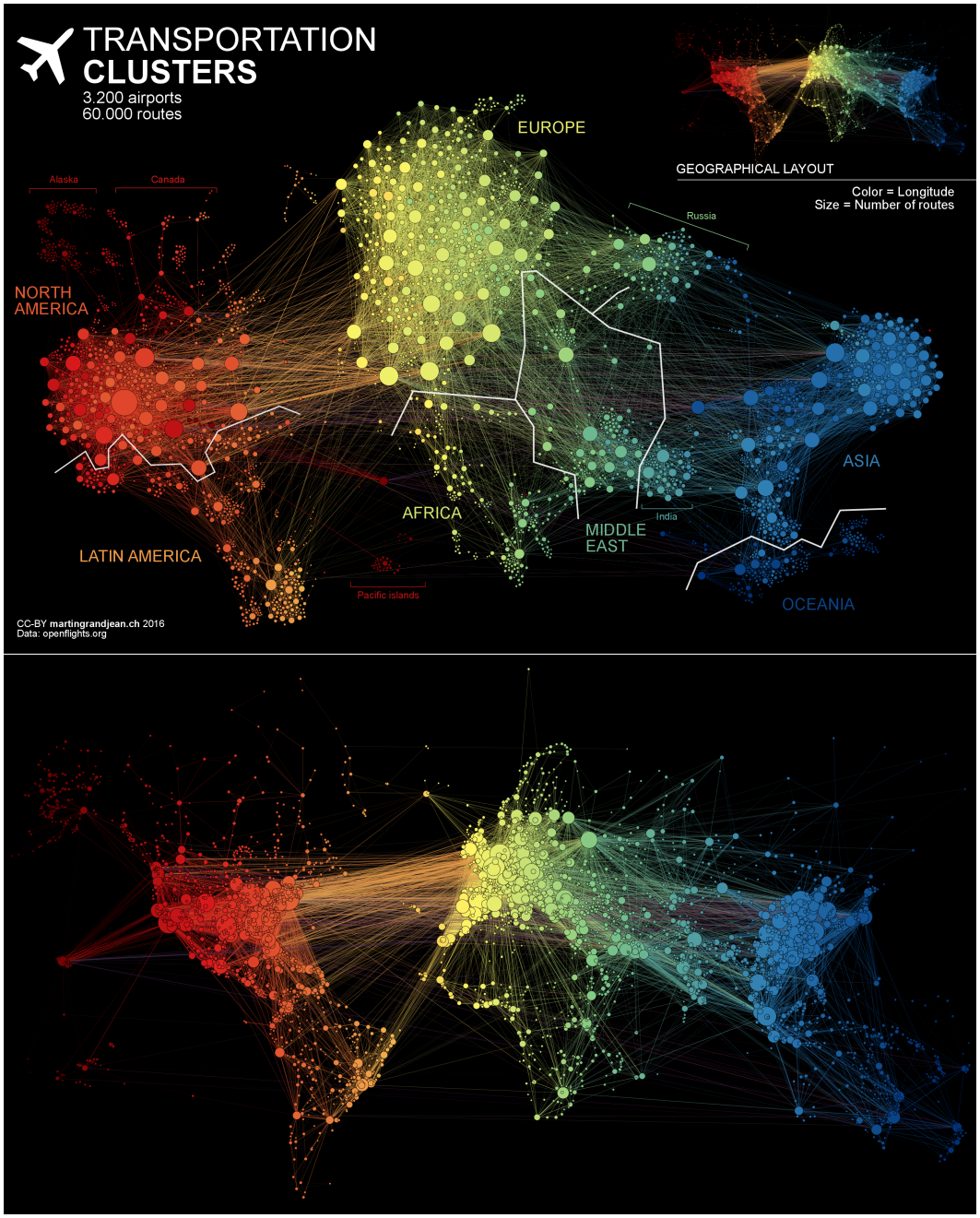


Figure 1.3: **Comparison of different representations of flight data by [Martin Grandjean, 2016].** The top figure shows the data represented by a force-directed graph layout (described below) and a Geo-layout showing each point at its location on the Earth.

1.0.6 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 1.0.7.3) 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].

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].

1.0.7 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 Table 2.4) 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 1.0.7.3), describing the merits of each layout.

1.0.7.1 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 1.4 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.

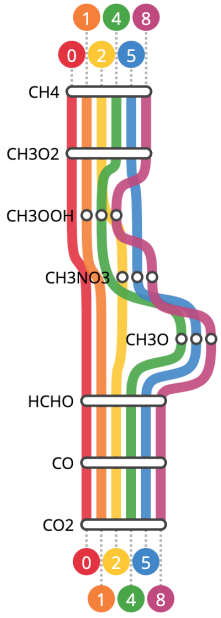
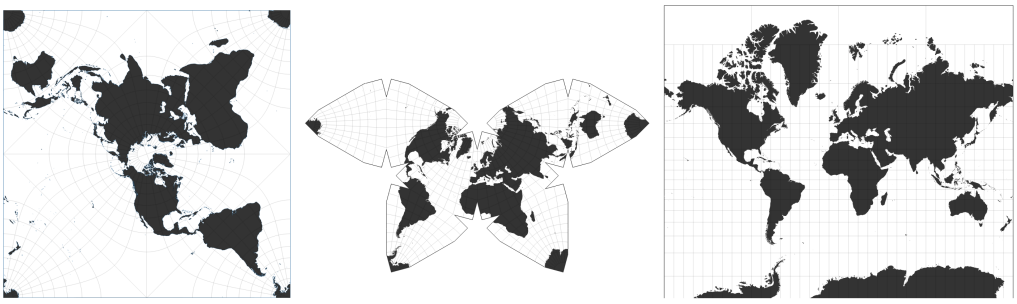


Figure 1.4: **A transit map showing all the possible routes from methane to carbon dioxide.** This was drawn using MemoryMap [Foo, 2019] and uses a version of the MCM methane subset, where carbon dioxide has been introduced.

1.0.7.2 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 1.5). 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).



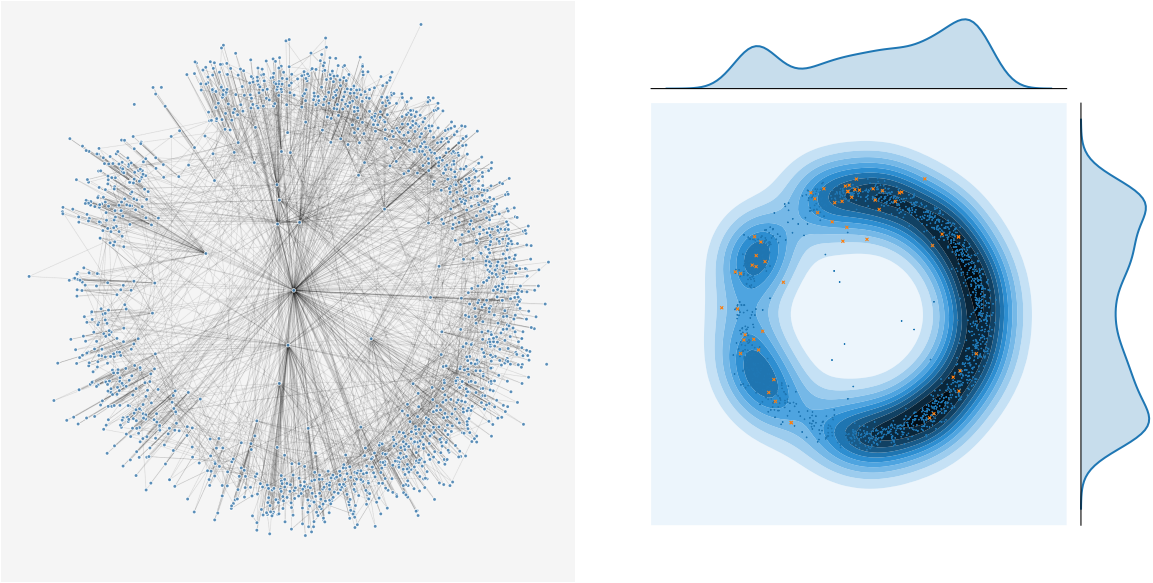
(a) Pierce Quincucial (b) Waterman Butterfly (c) Mercator

Figure 1.5: **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 1.6) where (a) shows the graph-based representation including links, and (b) shows the density distribution for all nodes. Figure 1.6b shows that primary emitted species (orange dots) are uniformly (radially) distributed for angles and Figure 1.6a 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

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

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 1.6: **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.

1.0.7.3 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].

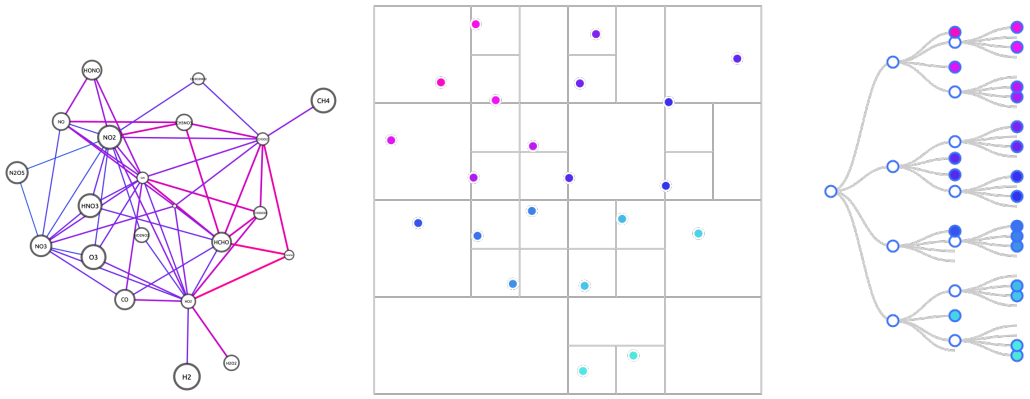
Next we describe the Barnes-Hut algorithm, a mapping algorithm which builds a hierarchical tree of the data by splitting a plane into quartiles. This is used within the many force-directed graph layouts, including those of Force Atlas 2 and Yifan Hu, described shortly. Once this has been done a selection 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 1.7.



(a) Methane Graph (b) Quadtree constructed from Figure 1.7a

Figure 1.7: **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 Subsection 4.0.11). 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.

1.0.7.4 Layout Selection

Subsection 1.0.6 explained the importance of removing overlapping edges and Subsection 1.0.6 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 1.0.7.4) and density (Figure 1.0.7.4) 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.

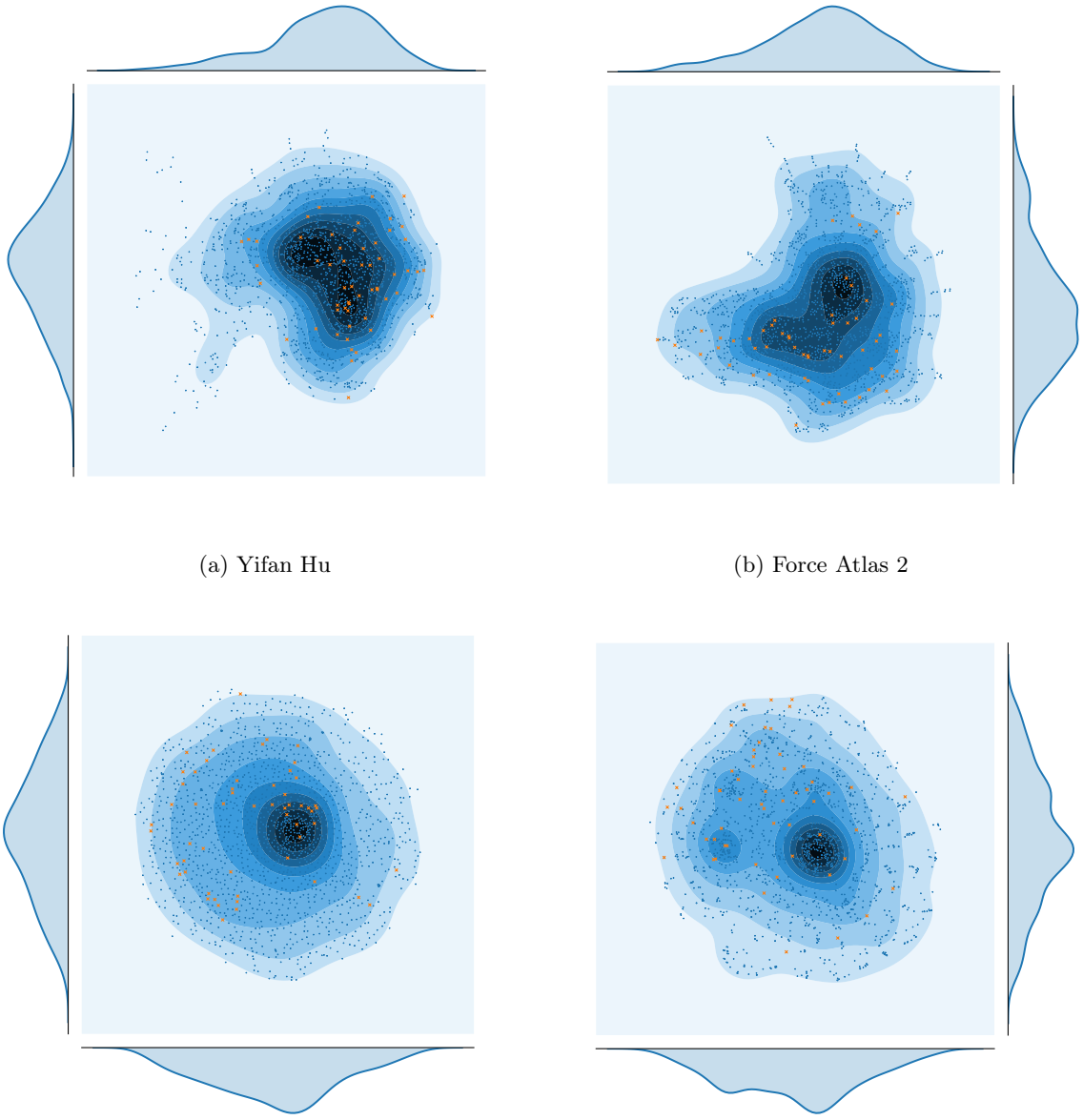
only 2% for orthogonal ones. In this subsection, I look at the use of different graph layouts, and their effect on the user readability of a graph.

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 1.6b and Figure 1.8. 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 1.6b), tsNet (Figure 1.8d) and Force Atlas (Figure 1.8b) 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 1.8a) and Mercator (Figure 1.6b) 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 1.8d) provides the best representation, followed by the OpenOrd and ForceAtlas layout.



(a) Yifan Hu

(b) Force Atlas 2

(c) OpenOrd

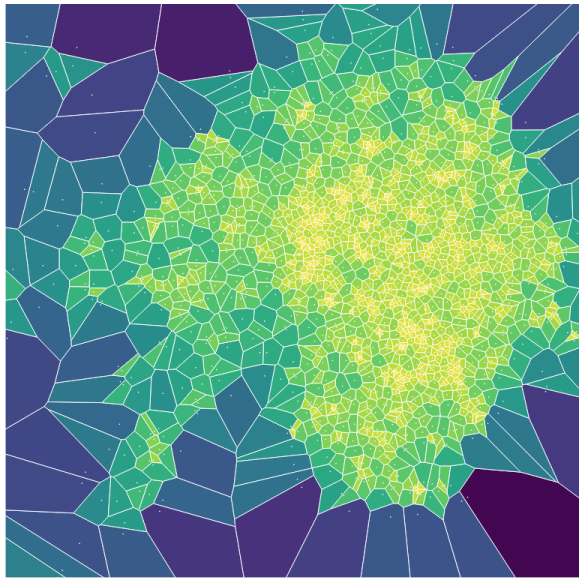
(d) tsNET

Figure 1.8

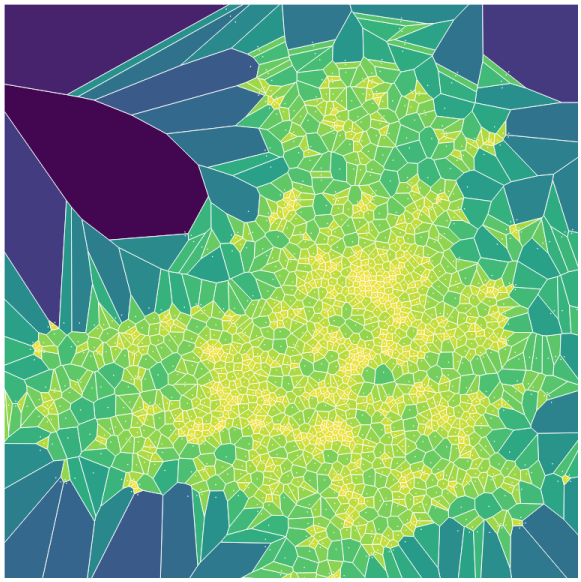
Node Density

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

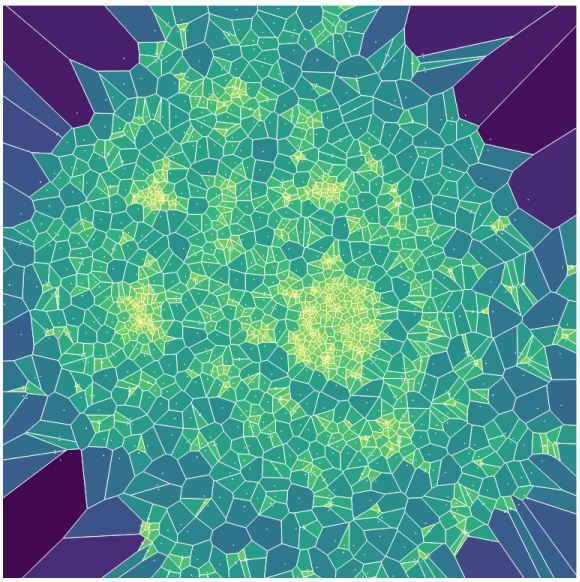
similar to a box full of bubbles, where each bubble fills the largest area it can before meeting another. Next, the area of each polygon is calculated and saved to produce a dataset representative of the complete density distribution between nodes. Here larger areas represents a species with distant 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 1.9. 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.



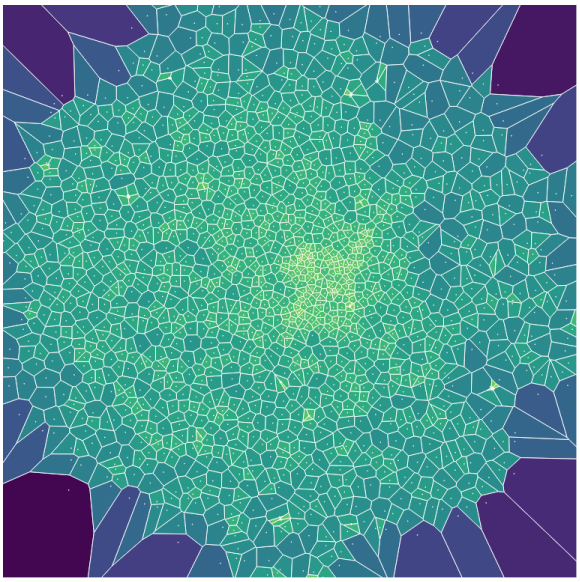
(a) Yifan Hu



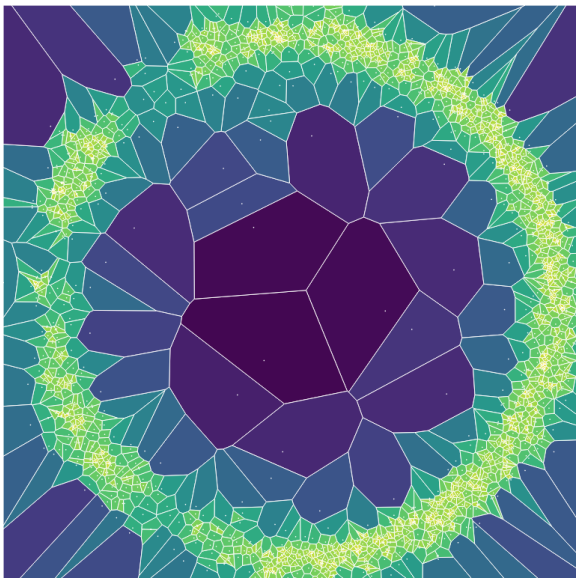
(b) Force Atlas 2



(c) tsNet



(d) OpenOrd



(e) Mercator

Figure 1.9: 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 1.10. 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).

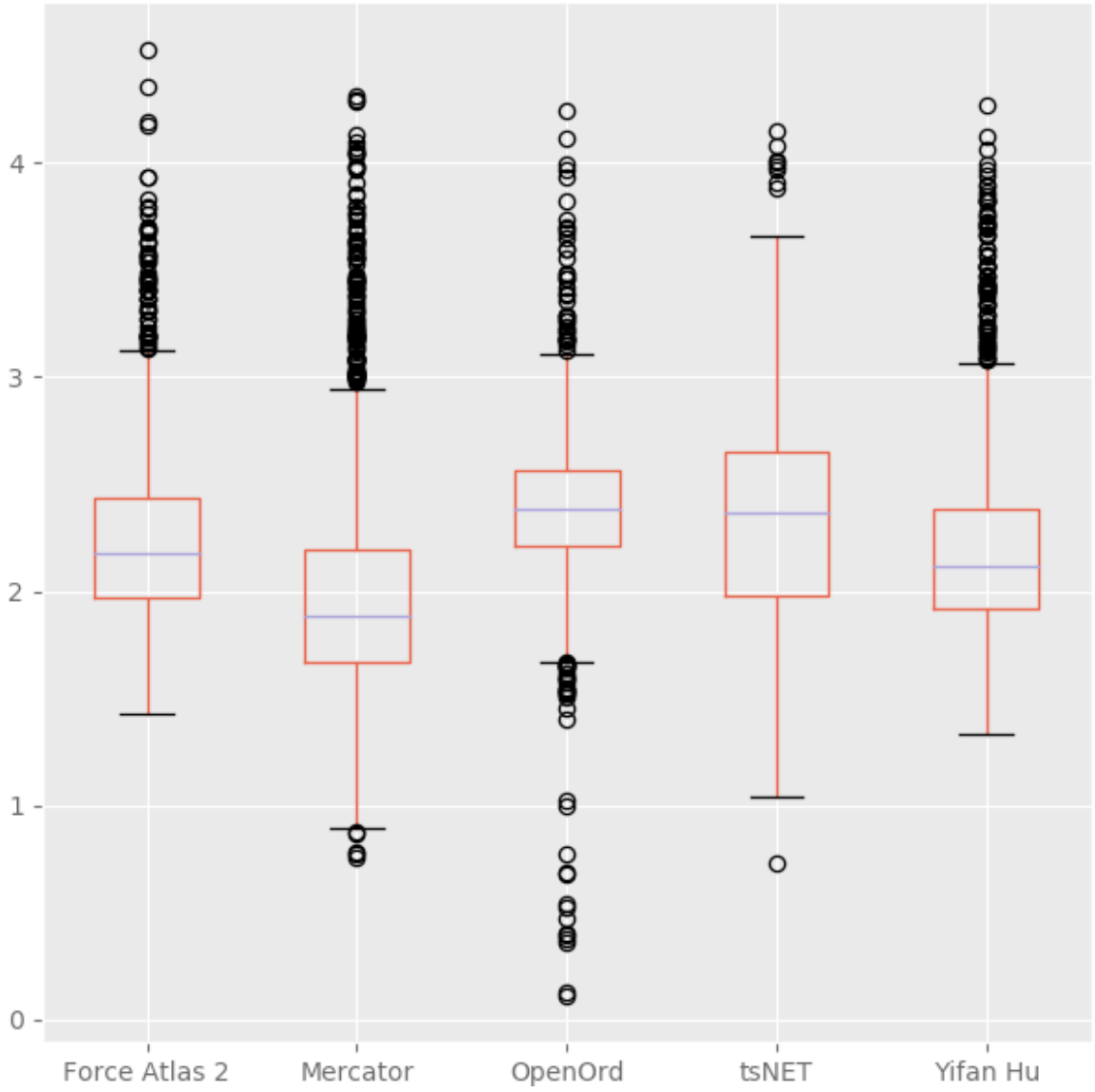


Figure 1.10: Voronoi $\log_{10}$(Area) BoxPlot for all plots in Figure 1.9

Applying these citerions to Figure 1.10, 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 1.0.7.4) makes it a better candidate than the Yifan Hu graph, which scored the highest in the boxplot test.

1.0.8 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.

1.0.8.1 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 1.11. 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].

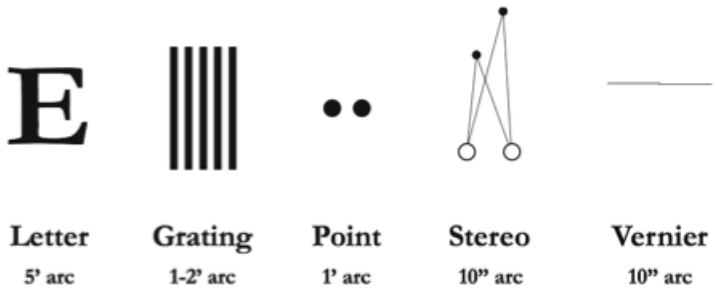


Figure 1.11: Important acuities in visualisation, Source: [Jankun-Kelly et al., 2014; Ware, 2013]

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.

1.0.8.2 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.

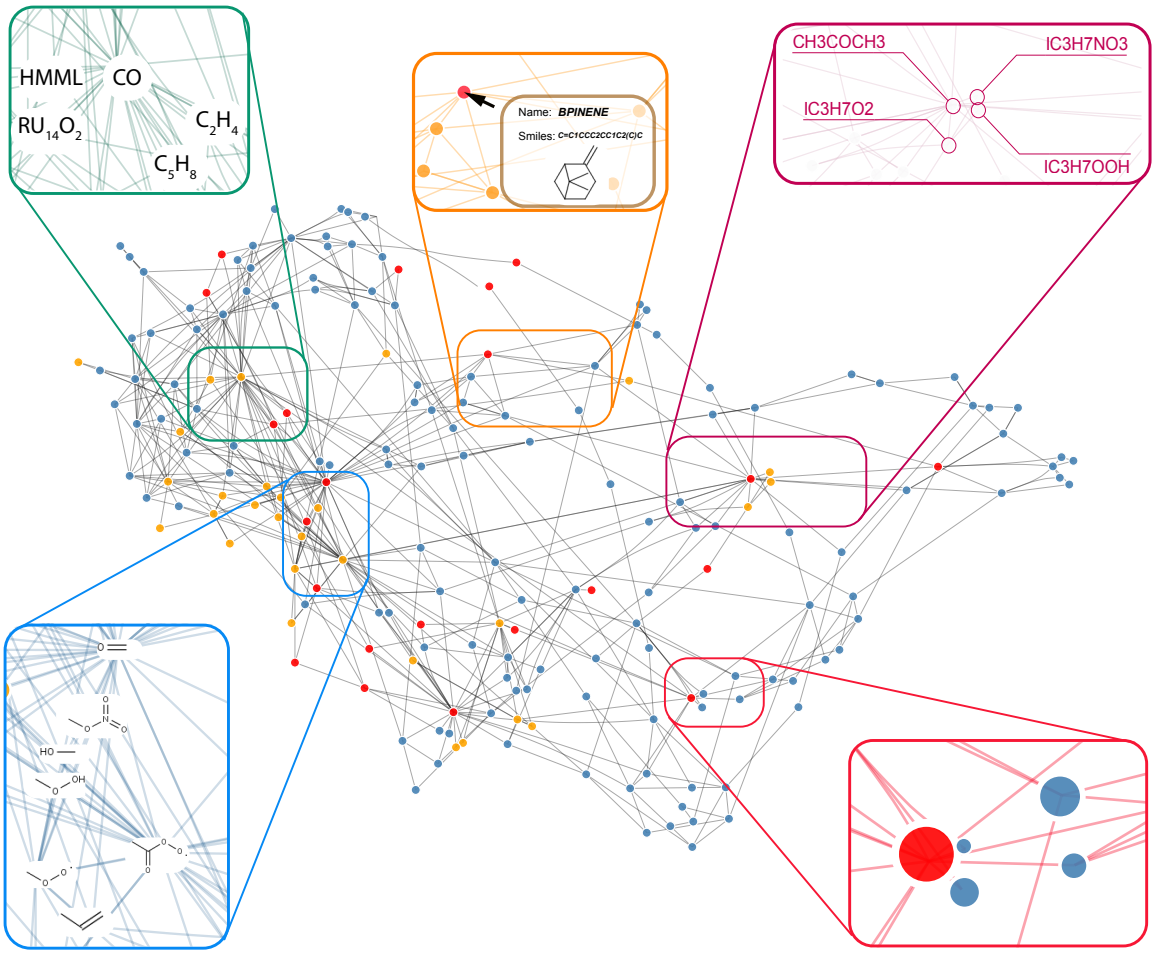


Figure 1.12: **A graph showing 5 different node encoding methods.** These are Circle Atrributes (red), Chemical structure (blue), Species Name (green), External Labels (maroon) and interactive selection (orange). The network shows the Common Representative Intermediate species [Jenkin et al., 2008] mechanism. 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 1.12 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.**

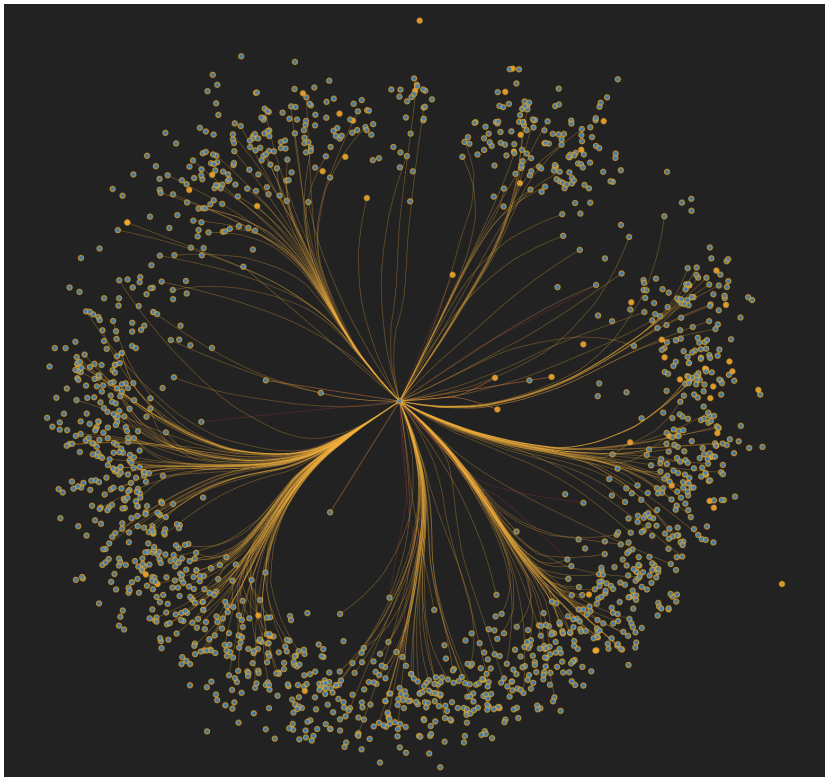


Figure 1.13: **Using mouseover edge-selection to hilight all links related to a node.** This figure shows how in using interactivity it is possible to reduce clutter and filter the information presented by a densely populated graph. In this case the mercator projection (Subsubsection 1.0.7.2) is used, 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].

1.0.8.3 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 1.14.. 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 1.14: **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.

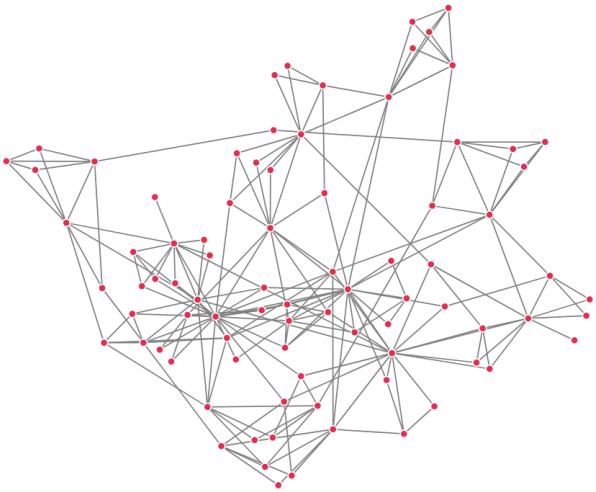
Edge Shape

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 1.15 shows a selection of different edge types for the Butane MCM subset.The liear network (Figure 1.15a) consists of straight lines betwen nodes.

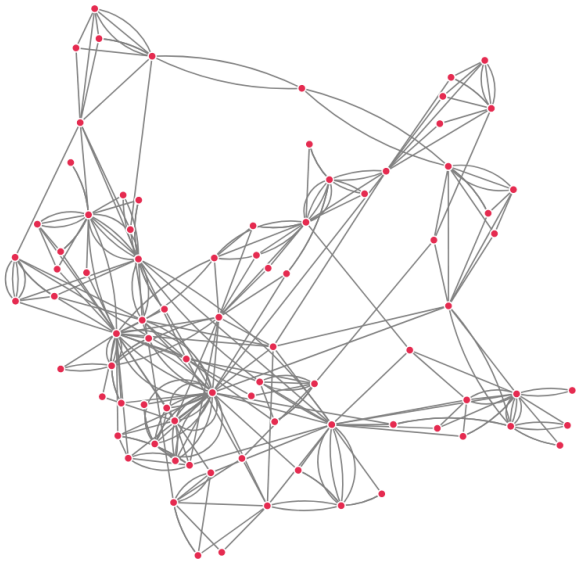
If a multi-edge graph is required, it is impossible to represent this as all edges between two nodes follow the same path. To improve on this a quadratic arc (Figure 1.15 b) can be used. This presents a symmetric representation where each edge is revealed. Finally bezier curves (described below) can be used to show an asymetric representation of the multi-edge graph (Figure 1.15c). 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)



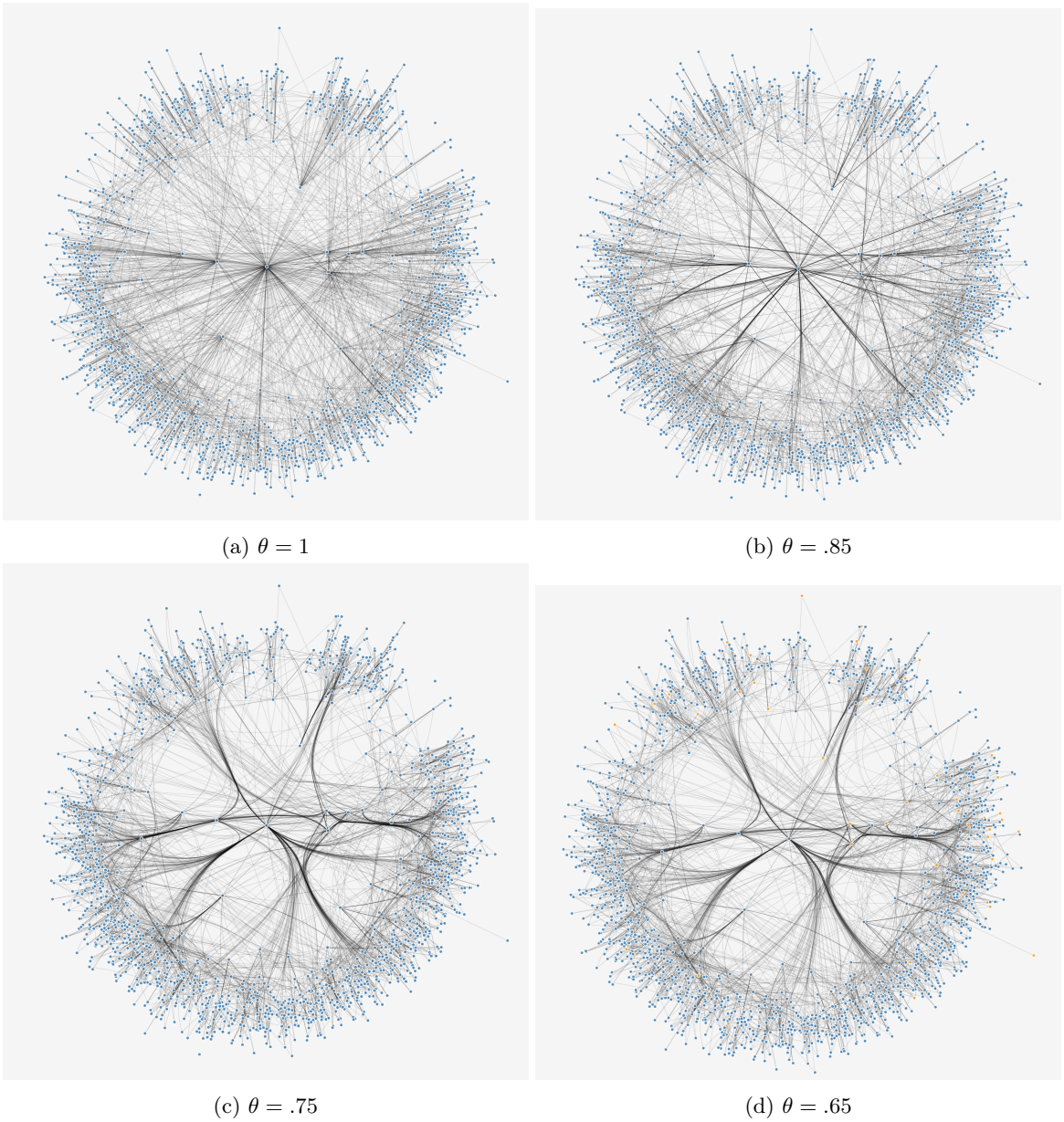
(b) Quadratic (multi-edge)

(c) Bezier (multi-edge)

Figure 1.15: **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 1.21 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 1.16: **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 1.17,[Bach, 2020].

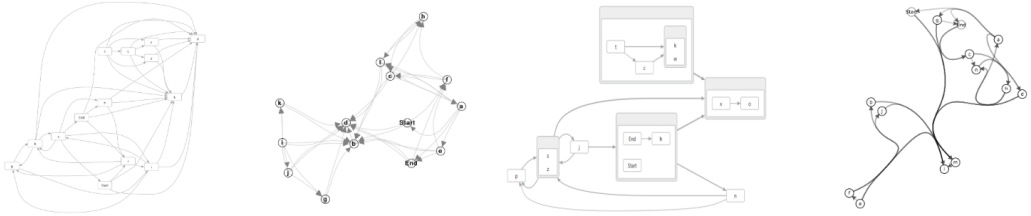


Figure 1.17: **An example of confluent bundling.** From left to right - A traditional network, Edge bundling, Power Graph and Confluent graph representations. Source: [Bach, 2020]

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 1.18. 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 1.19. Finally crossing links are removed, leaving the confluent graph, Figure 1.20.

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

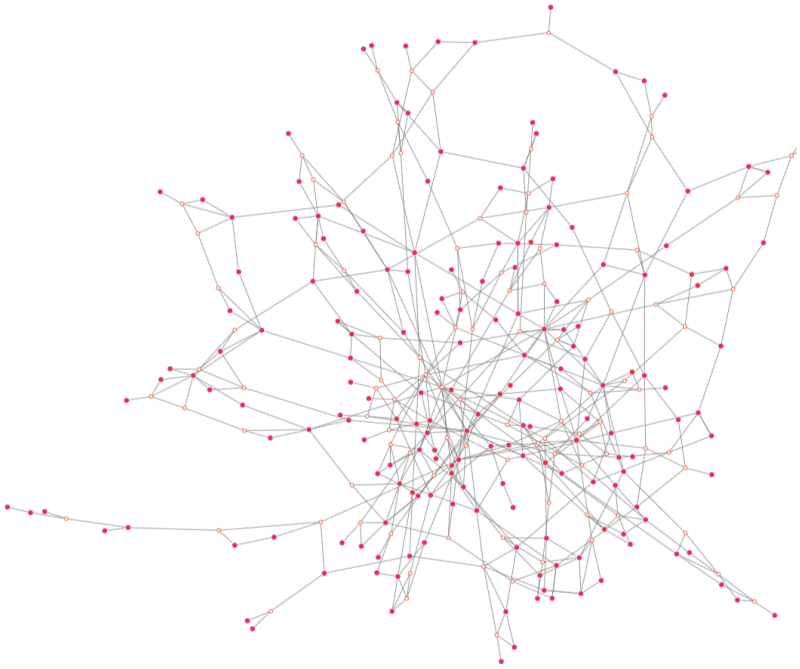


Figure 1.18: **The routing graph of the butane mechanism.** Here paths which contain two or more bundles have an extra 'routing' node introduced (orange stroke)

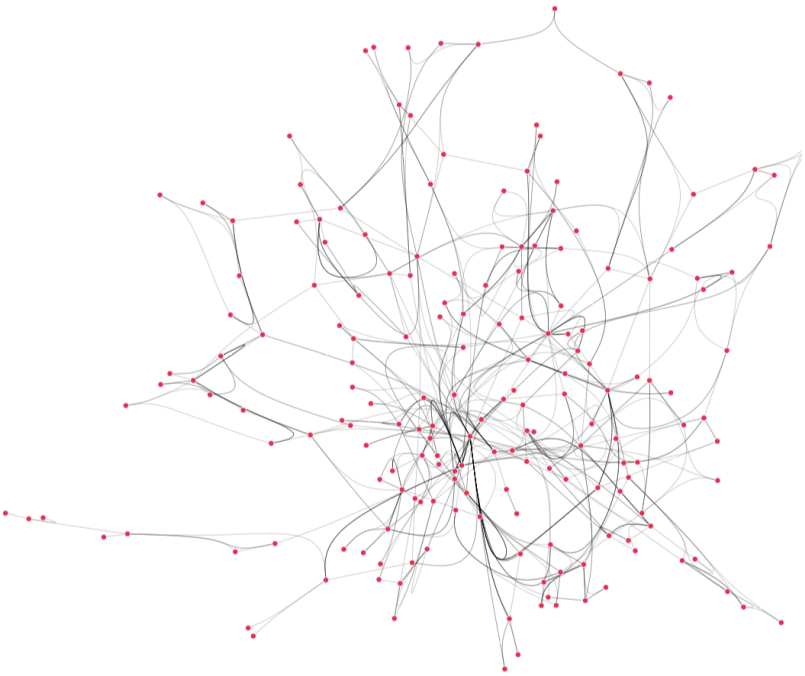


Figure 1.19: **Confluent graph with crossing artifacts.** The routing graph with the addition of basis-splines using the orange routing nodes in Figure 1.18 as control points.

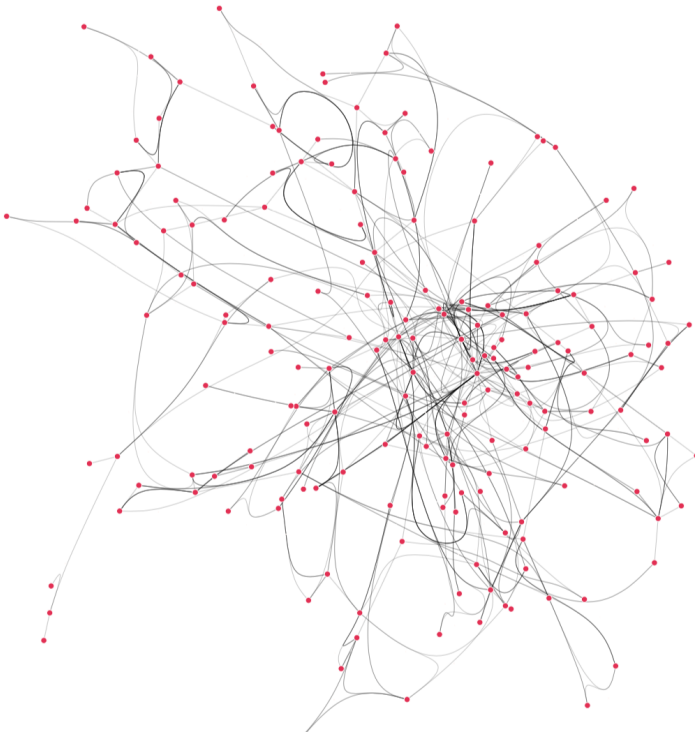


Figure 1.20: **Confluent graphs without crossing artifacts.** The remaining confluent graph with crossing edges removed.

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].

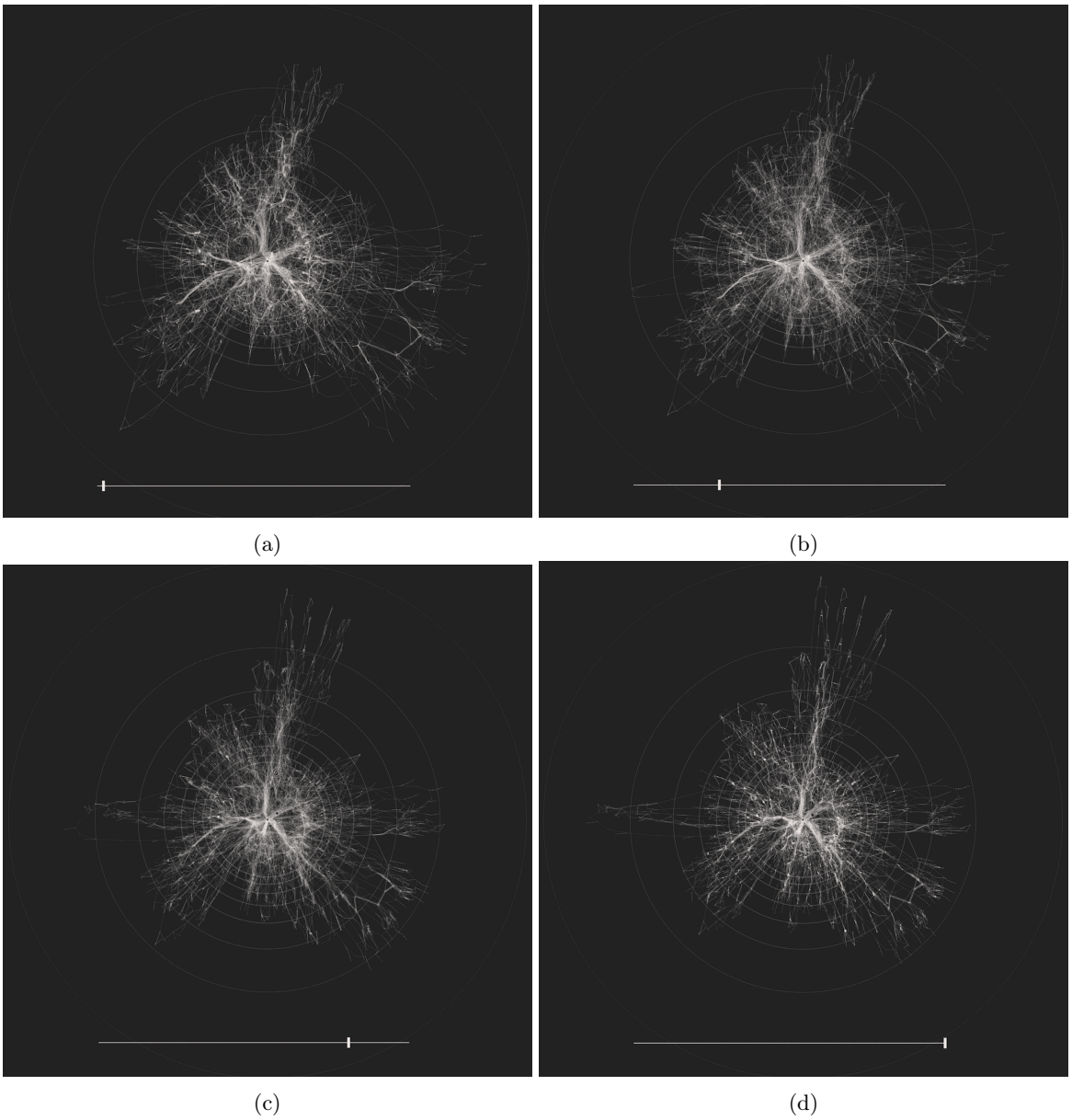
Reducing the angle between related edges increases readability and allows the behavioural process to infer information about a graph correctly. This process can be compared to predicting the direction of turbulent vs laminar flow. In addition to this edges should be spaced evenly around node, maximising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

1.0.9 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 **??**.



(a) (b)

(c) (d)

Figure 1.21: **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.

1.0.10 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.

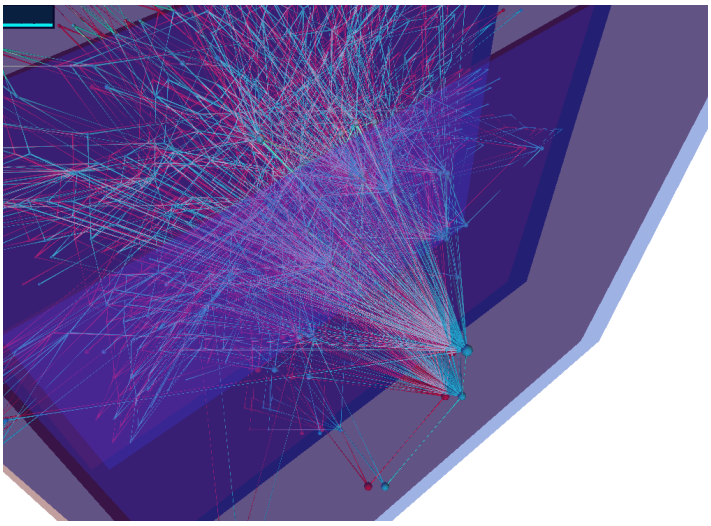


Figure 1.22: **A 3D representation of a graph to hilight certain features.** The first, second and third generation species of isoprene shown as an interactive 3D anaglyph.

1.0.11 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.

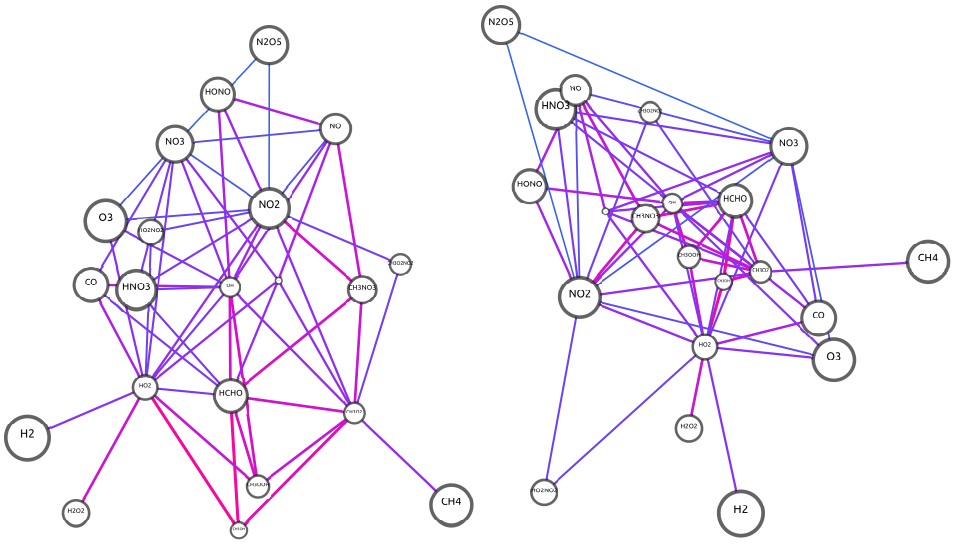
This allows a simplified view of the different properties which affect the visualisation of the graph produced.

1.0.12 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.

1.0.13 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 1.23: 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

1.0.14 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 1.24. 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 1.25

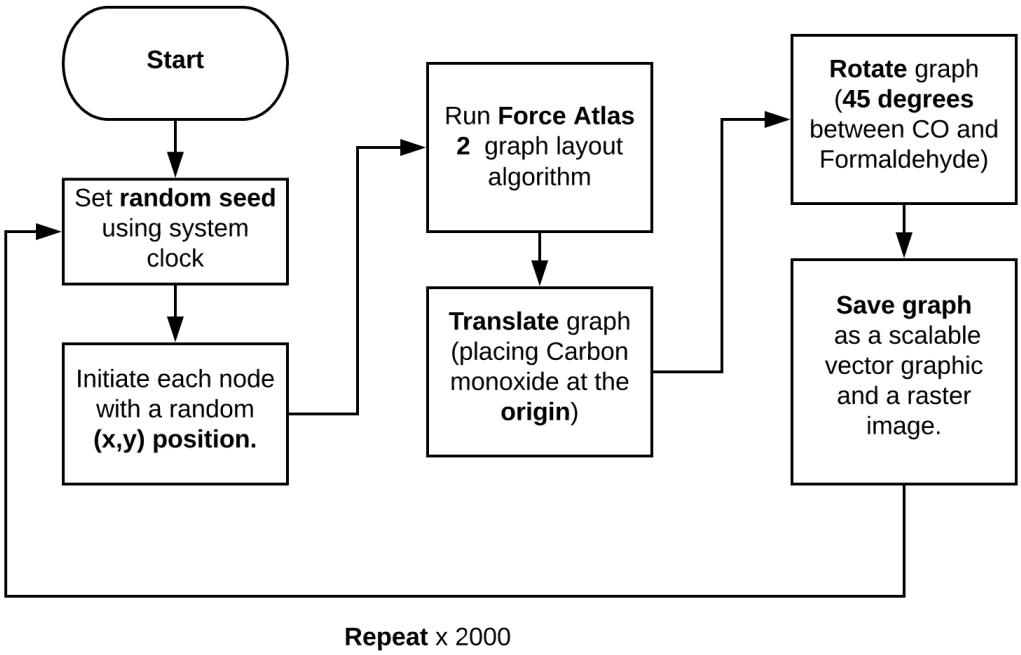


Figure 1.24: A flow chart of the process performed by the custom gephi script used to generate the data set

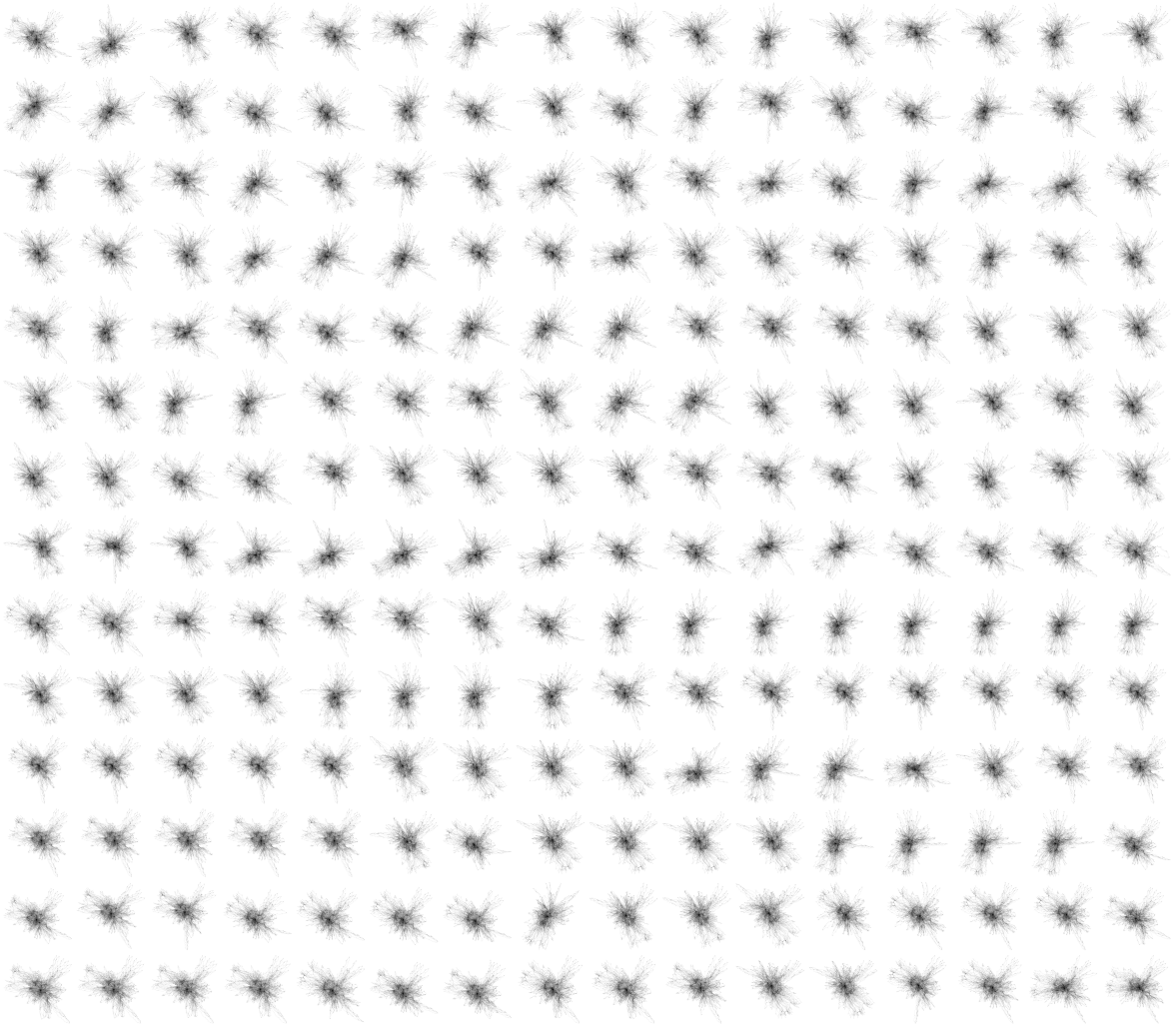


Figure 1.25: A sample of the 900 graphs generated using the force atlas 2 algorithm for the simulation output representative of the summer beijing chemistry at noon.

1.0.14.1 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 1.25 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 1.26 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.

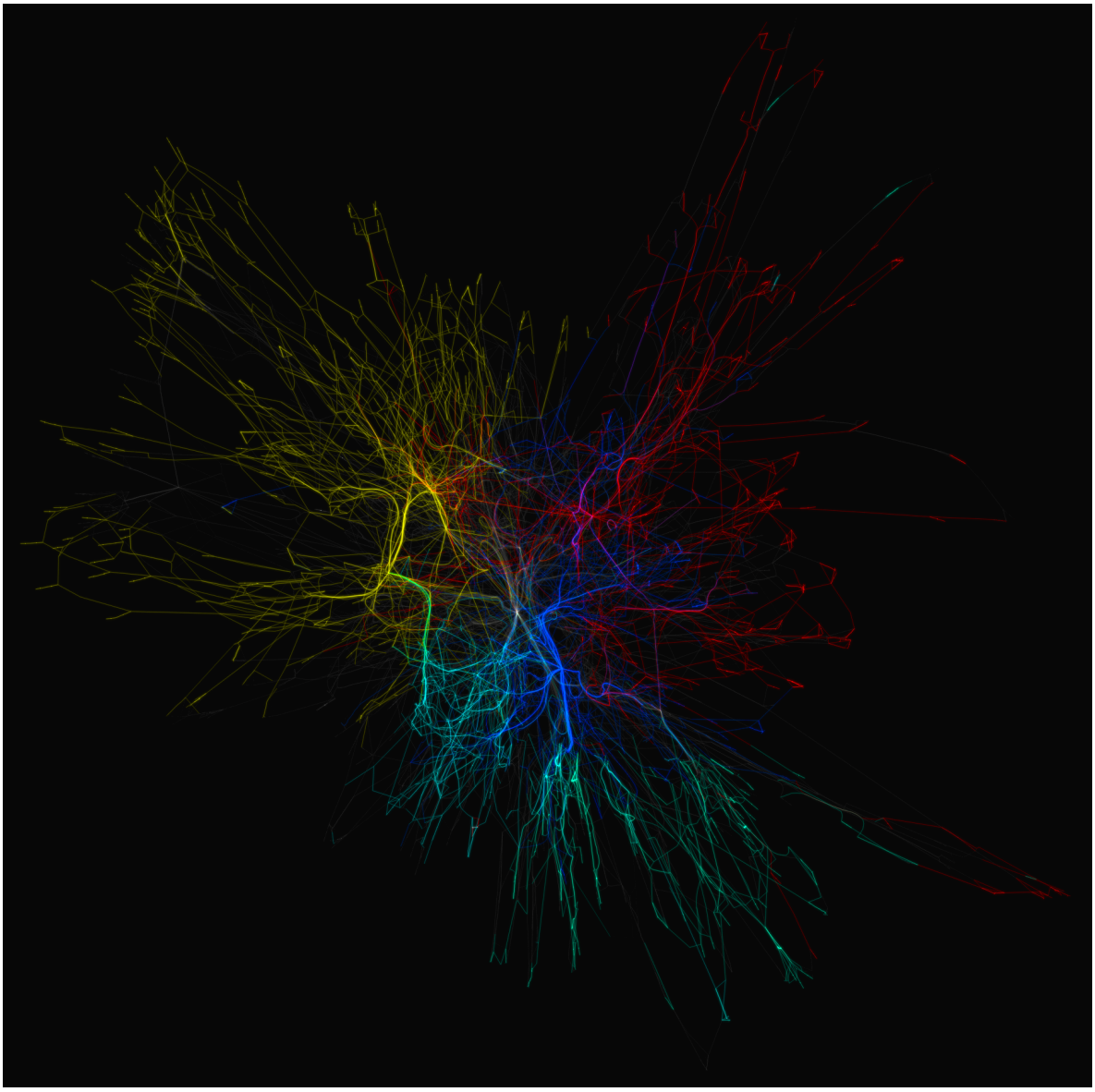


Figure 1.26: Hi-lighting the groups of species, and their products within the MCM network graph. These are Aromatics (gold) , terpenes (turquoise) and Alkane/Alkene carbon chains (red/blue)

[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

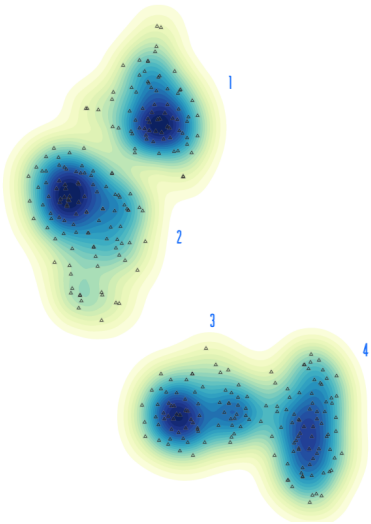
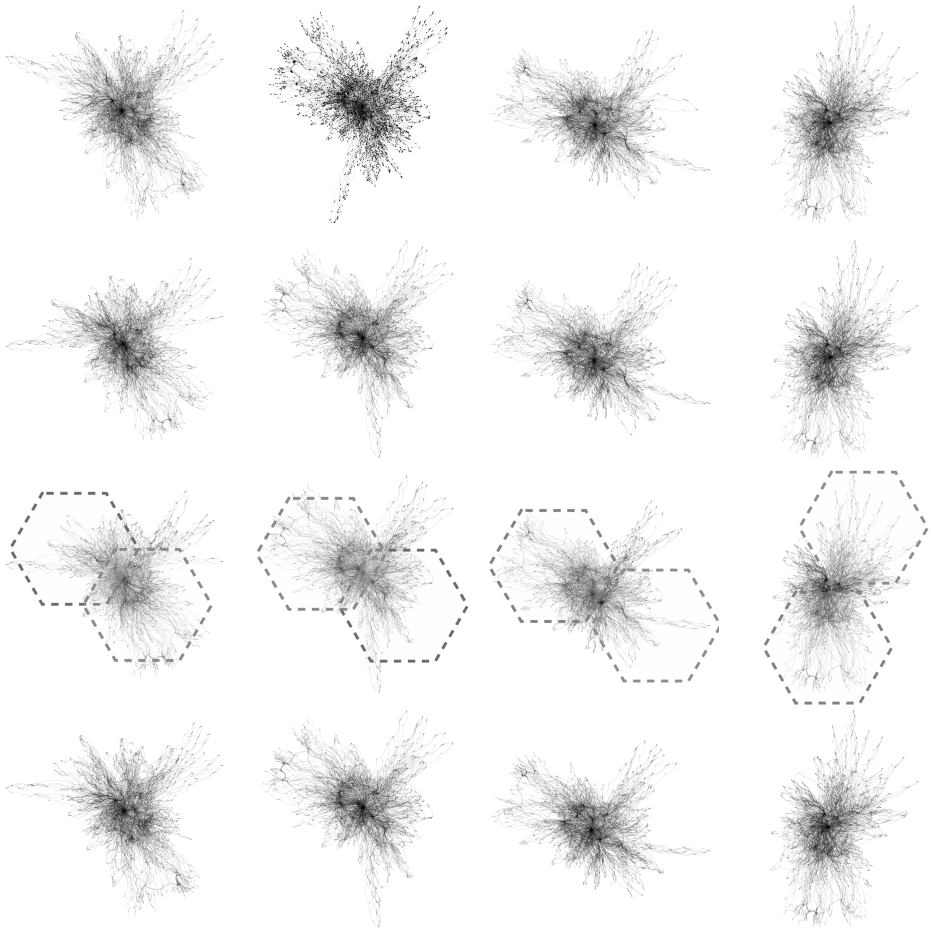


Figure 1.27: **A normalised scatter plot of 2D space produced by the t-SNE algorithm.** Each triangle represents a different file, and the colours/density contours show the regions in which we find similar images/graphs.

Using interactivity and/or vector cluster detection techniques it is possible examine which files contribute to an area of high density. Figure 1.28 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 1.28: **A selection of graphs corresponding to the labeled clusters in Figure 1.27**. These reveal that symmetric similarity between like-positioned points within the t-SNE output.

1.0.15 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.

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.

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.

Chapter 2

Chemical model diagnostics using graph theory and metrics.

"The complexities of cause and effect defy analysis."

- Douglas Adams, *Dirk Gently's Holistic Detective Agency*

2.0.1 Introduction

The node-link (ball-stick) [REF SECTION] style structure has long been used to represent real-world relationships between items. Such a structure is complementary to our cognitive disposition towards pattern recognition [citep]. It is for this reason that the node-link visualisation format has been used for anything ranging from transportation maps [citep BECK] to the differentiation of ancestorial lineages of the human race (Figure 2.1). However, the abundance and complexity of real-world data often present us with difficulties in manually representing it in a useful form. In SECTION XX it is suggested this may be overcome with the use of computational analysis and automated visualisation tools. Such methods usually require a level of data manipulation to transform the data into a machine parseable form.

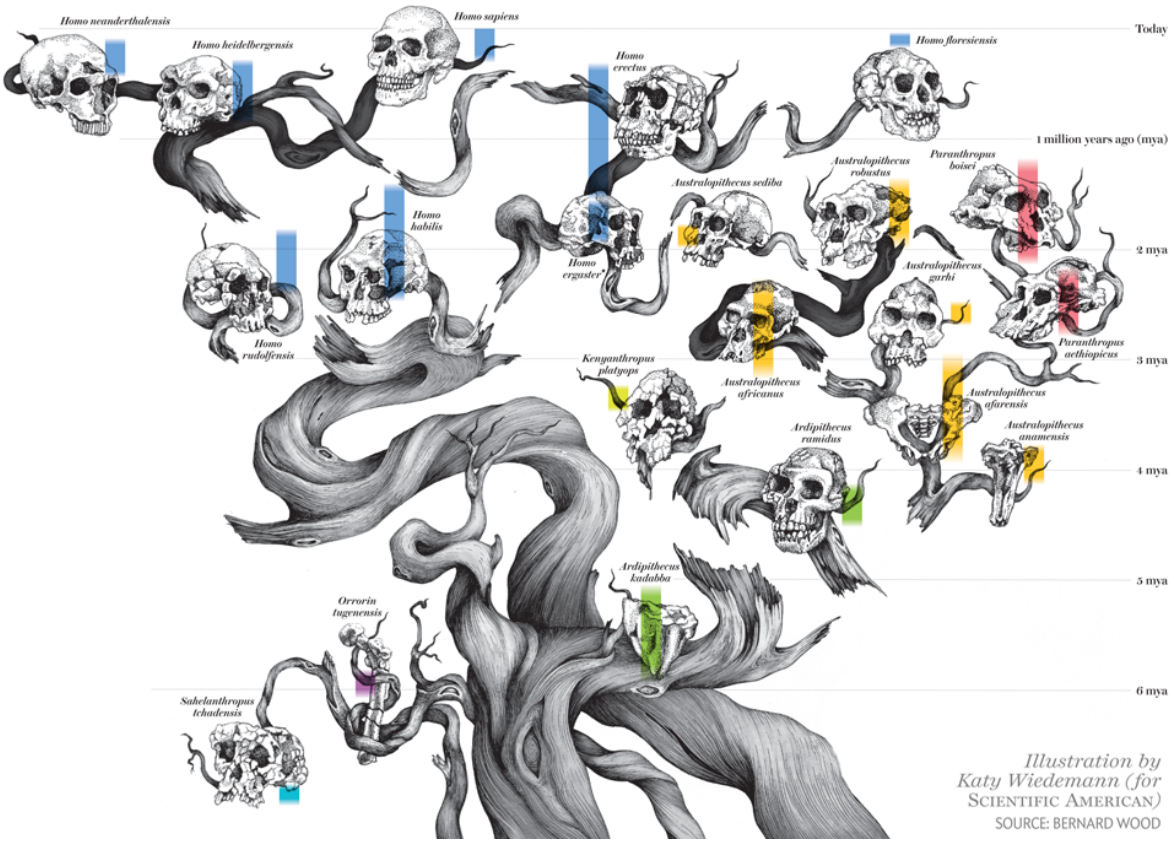


Figure 2.1: **The human family tree.** This is a visual depiction of the human lineage, starting with our common ancestorial roots. In SECTION it was shown that the use of trees / graphs[1]is useful in showing relationships between items. Source: [Wood, 2014]

In the field of mathematics a graph, $G(\nu, \epsilon, \omega)$, is defined as a function of items (vertices[2]), ν which are connected through a series of connections (or edges[1]) representing any relationships between them, ϵ.

[1]A tree is a special case of a graph

[2]The term node, item or vertex shall be used interchangeably for the remainder of this chapter. This also applies to links/relationships/edges and edge-weight/strength

Since relationships in the real world are rarely equivalent, we then encode the importance of each link in the form of an edge weight, or strength - ω. Such formats allow both numerical and computational algorithms to understand and interpret the graph structure, providing us with information about the data or make use of automated layout programs for visualisation.

This chapter builds on the work shown in SECTION XXX - where the ability to represent complex data in the form of a graph was used to (visually) draw information regarding network structure and temporal changes. Here I will begin by exploring situations where the visual representation of many, large or complex networks is impractical. We start by introducing a series of mathematical approaches which are capable of quantifying the graph (and nodes within it) and apply them to the co-author network for papers regarding the Master Chemical Mechanism, Subsection 2.0.2. Following these global metrics are used to categorise the chemistry within different mechanism subsets, and provide us with an insight to the chemistry structure (SECT LABEL) and finally apply these to real-world simulations representing a range of environments (marine, rainforest and urban) in SECTREF.

This allows for a higher level of automated analysis which can be used to batch process, analyse and categorise chemical simulations. Subsection 2.0.2 begins by introducing the most common of the graph metrics which can be used for analysis. To do this a citation graph is generated by web-scraping google scholar results.

2.0.2 Graph Metrics

An increase in the ability to gather and store data results in a difficulty to understand it (ref SECTION). The production of large, multivariate networks of inexplicable complexity greatly hinders our ability to draw out meaningful conclusions based on visualisation alone. This means that much like the generation of mechanism, or creating semi-automated graph drawing layouts, we must rely on the field of mathematics coupled with computational aid (REF SECTION).

Numerical algorithm, derived from the field of Graph Theory can be used to circumvent the need for individual graph analysis and provide us with information about the network. One such subset of numerical algorithms are regarded as "centrality metrics", and may be used to rank the role and importance (centrality) of a node. In the following sub-section, the most common (REF PAPER) centrality metrics are discussed and applied to the MCM citation network.

2.0.3 Centrality Metrics And Academic Publishing.

One common application for graph analysis and visualisation is the representation and prediction of citation counts within academic journals [Small, 1973; Page et al., 1999; Monastersky and Van Noor-

den, 2019; Molontay and Nagy, 2020]. Here network-visualisation techniques may be used to highlight the origins of a paper - for instance, Figure 2.2 shows the multi-disciplinary research which underpins 6 prominent discoveries in the last 150 years.

To the properties presented by different centrality metrics (described above), we apply them to an approximate representation of the citation graph relating to the Master Chemical Mechanism (Subsection 2.0.4).

2.0.4 The Master Chemical Mechanism (MCM)

The MCM, [**?**], is a near explicit representation of our foremost understanding of gas-phase tropospheric chemistry. The mechanism describes the oxidation of 143 primary emitted VOCs and the respective rates at which this occurs. It has been used in the...

Information on the chemistry, - x species - y ... first published and how this can be used with regards to the following algorithms are presented in REF JENKINS 15 ACP.

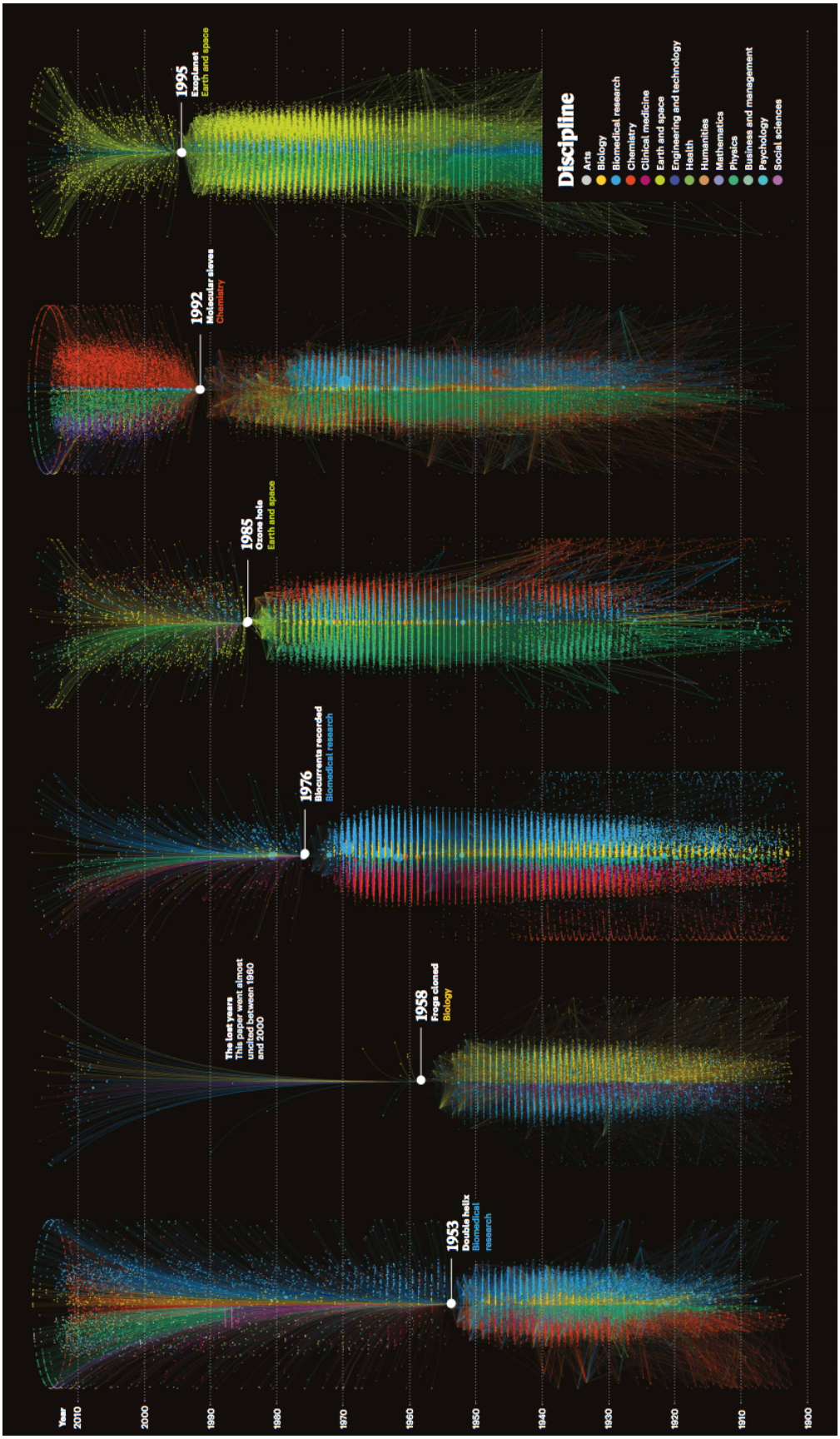


Figure 2.2: **150 years of letters to Nature.** A visualisation showing how previous research is used to inspire future studies. Important discoveries (DNA, Cloning(frogs), Bio-Currents, Ozone Hole, Molecular Sieves and Exoplanets) are split into research which contributed to their formation (below), and the consequent papers produced from each discovery. Use of colour is used to emphasise the multi-disciplinary nature of prolific scientific discovery. Source: [Barabási, 2019]

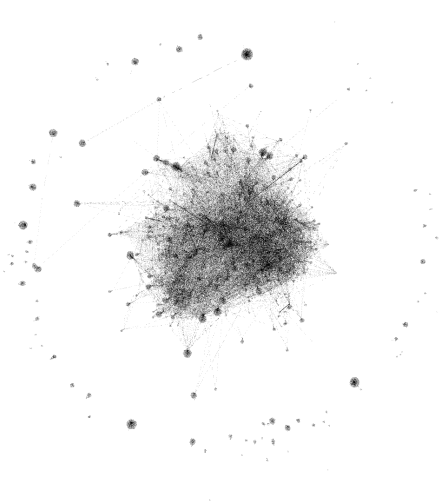
2.0.5 Data Collection

To generate a dataset on papers related to the MCM. The academic search engine (Google Scholar [Google, 2019]) is queried for all articles containing the words { *"Master", "Chemical", "Mechanism"* and *"MCM"* }. For each match, the first 100 pages of results are selected. Each of these contains 10 articles, from which the first 100 pages of related articles are chosen. In taking the top 1000 citations for each page a network of 15744 papers and 30178 citations[3] is created. This process made use of an edited version of the *etudier* Github repository, [Edsu and Ellis, 2019].

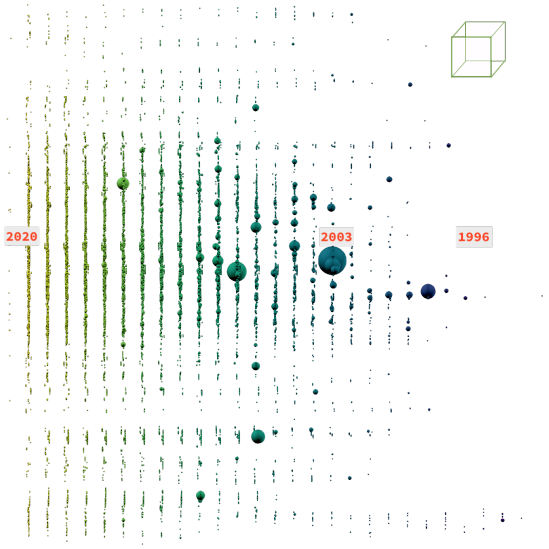
2.0.6 Visualising The Data.

The initial visualisation of the dataset is accomplished through the use of THREE.js [Cabello, 2019]. This makes use of WebGL bindings and allows for the efficient viewing, querying and interacting of the data in 3 dimensions. This helped identify the temporal changes within the network by mapping a papers publication year to the z direction, Figure 2.3, as discussed in Subsection 2.0.7.

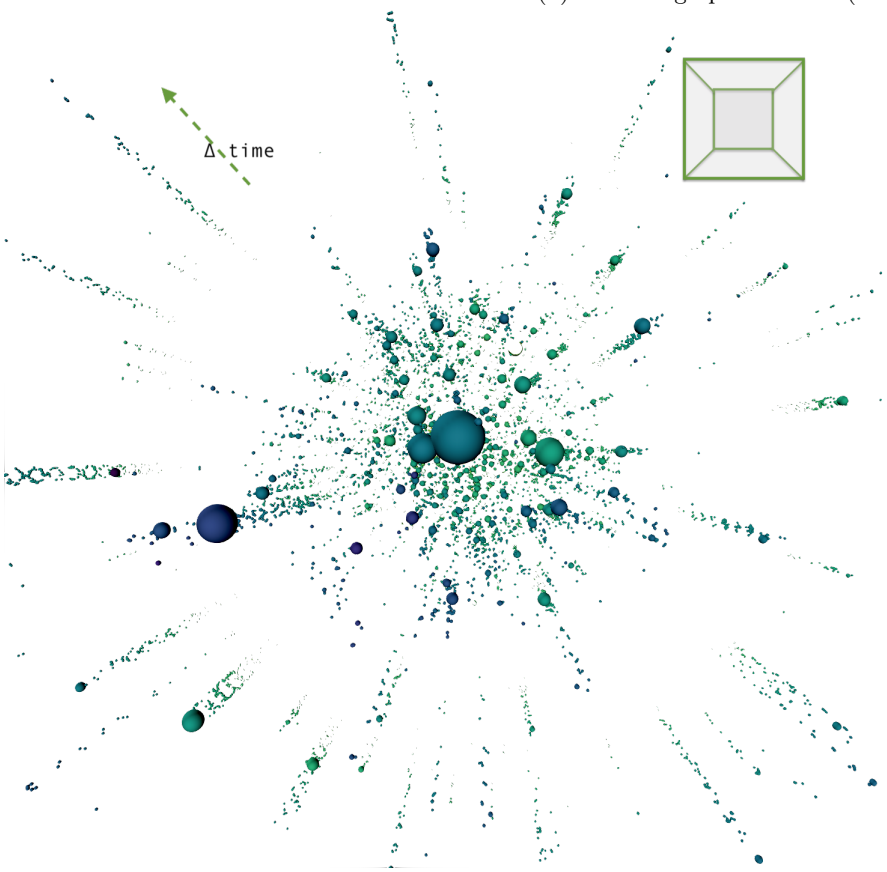
[3]Note: this had the potential of returning up to 1000,000 nodes



(a) A 2D force directed representation of the network using the gephi software [Bastian et al., 2009]



(b) 3D orthographic camera (sideview)



(c) 3D perspective camera

Figure 2.3: **Initial 3D graph representation of the scraped MCM citation graph.** (a) shows the 'classic' graph representation of the network. (b) shows a size representation using an orthographic perspective. Here time is shown across the x axis, with yellow being the most recent. (c) uses a perspective camera, which emphasizes the... Still captures of 2D and 3D visualisations of the dataset. Node size corresponds to the number of citations, and colour (and z-axis) corresponds to the publication year for each paper.

2.0.7 Filtering The Data

In the method used to web scrape data, there are several features which need to be corrected/removed. The reasons for this are discussed below.

Pre-1996

There exist several papers predating the conception of the MCM (1996). A number of these can be attributed as incorrect data, with publication dates <1900 which may be the result of missing information or a fault in googles web scraping algorithm. Any such papers are removed from the dataset.

For otherwise correct articles, those published pre-1996 are also filtered from the dataset - this is because we are interested in identifying the influence the MCM has had on research and not the research that may have led to its creation. This can be seen in the cone-like shape emanating from the first MCM papers in Figure 2.3b.

N-th degree research

Not all research articles in a field reference other articles with the same field. Figure 2.2 showed us that many of the great discoveries in science have a multidisciplinary nature. It is for this reason that it is expected that articles from non-atmospheric areas of research may reference or build upon specific areas of research touched by the MCM. Such papers, and in consequence the papers which cite them, have little or no links to many of the core MCM papers. Such papers manifest themselves as a halo of satellite clusters which are connected by themselves but not with the main body of the graph, Figure 2.3a. In using a 3D perspective viewpoint (Figure 2.3c) it is possible to identify the paper which references the MCM and then the consequent papers which cite it by observing the satellite clusters, and the gradually lightening spiral of papers which emanate out of it.

Analysis of the network connections for each cluster can allow us to identify the indirect relationships some of these diverse topics (Table 2.1) contained within the satellite nodes. Here it can be seen that the use of photochemical ozone creation potentials [Derwent et al., 1998; Jenkin and Hayman, 1999] are used for the Life cycle assessment of Italian high-quality milk production [Fantin et al., 2012]. Similarly indirect paths such as the paper: "Temporal controls on dissolved organic matter and lignin biogeochemistry in a pristine tropical river" ([Spencer et al., 2010]) can be used to link to [Stubbins et al., 2008] and ultimately the MCM protocol paper [Saunders et al., 2003].

If we desired to remove such papers, the simplest method would be to recreate the graph into one

where links are drawn between papers that are cited together (Subsection 2.0.8) and then removing any nodes without any external connections (isolates).

Fabrication of Bioinspired Actuated Nanostructures with Arbitrary Geometry and Stiffness	[Pokroy et al., 2009]
Temporal controls on dissolved organic matter and lignin biogeochemistry in a pristine tropical river	[Spencer et al., 2010]
Neuroproteomics in Neurotrauma	[Ottens et al., 2006]
Fast start-up of a pilot-scale deammonification sequencing batch reactor from an activated sludge inoculum	[Jeanningros et al., 2010]
Red blood cell oxidative stress impairs oxygen delivery and induces red blood cell aging	[Mohanty et al., 2014]
Life cycle assessment of Italian high quality milk production.	[Fantin et al., 2012]

Table 2.1: A selection of research papers not directly connected to the field of atmospheric modelling.

Unprobable occurances

Finally, the extracted network also contains many disconnected component subgraphs - graphs with no connection to atmospheric science. An example of this is seen in an article about neuroproteomics in neurotrauma [Ottens et al., 2006]. In analysing the paths which connect this, it is seen to cite the paper on "Large scale gene expression profiling of metabolic shift of mammalian cells in culture", [Korke et al., 2004]. This is an anomaly which within its structure contains the words "Master", "Chemical" and "Mechanism" (separately) and has ‘MCM’ as an abbreviation for one of the author names. To remove such papers, all disconnected sub-components are removed from the analysis.

A note on unintentional filtering

Author names and some extended titles may be truncated with the use of ellipses. This is due to the web scraping script extracting these directly from the Google scholar page, and not the original articles themselves. It is worth noting that the results in this section are not explicit, but rather a demonstration of graph theory on a real-world dataset.

2.0.8 The Co-Citation Network

The document coupling techniques of co-citation was introduced in the 1970s as an alternative approach for quantifying the results within the science citation index [Small, 1973]. Rather than rep-

resenting a graph using backpropagation (through the use of referencing and citation counts), a co-citation network introduces a link between papers if, and only if, they have been cited together. Although this loses the directionality of a graph, it allows us to show forward propagating trends between papers within the same field.

Applying the above method allows us to reduce the citation graph of 451 papers and 5402 edges to an undirected co-citation graph of 2758 edges - halving the number of original links between papers.

2.0.9 The Co-Authorship Network

An alternative to exploring which papers which are cited together are to look at their authors. Here undirected links are drawn between authors on the same paper. This style of analysis was used to show that the number of papers per author, and the total number of authors per paper can vary between research fields, [Newman, 2004]. In combining this with a series of network centrality metrics, [Fujita et al., 2017] revealed that it is possible to discern promising researchers from both iter and Intra disciplinary groups.

In building a co-authorship network for the MCM, we can identify authors who publish together[4] and highlight research groups who work with the MCM, Figure 2.4. This shows how authors with a similar geographic location/institution are more likely to publish together. The largest cluster here falls under the MCM developer team, which resides between the Leeds and York universities. Next two German institutions which are heavily involved in the atmospheric chemistry field (Julrich and Max Planck), followed by an assortment of Chinese authors, mainly centred around the Beijing or Hong Kong region.

[4]Disclaimer: as mentioned earlier, not all authors for every paper were recorded by the web scraping algorithm

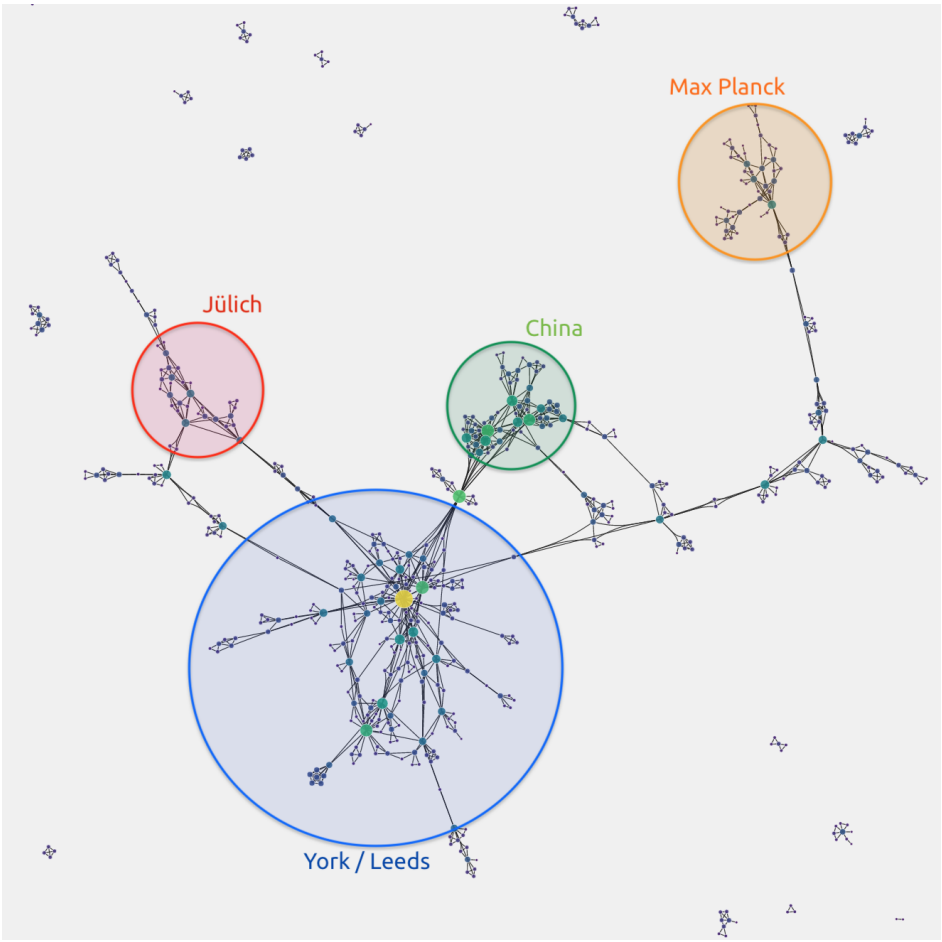


Figure 2.4: **The co-author network.** In representing the authorship network as a force directed graph we are able to see cliques or clusters of people who publish together. It can be noted that this often occurs when they have a similar geographical location.

2.0.10 Metric analysis

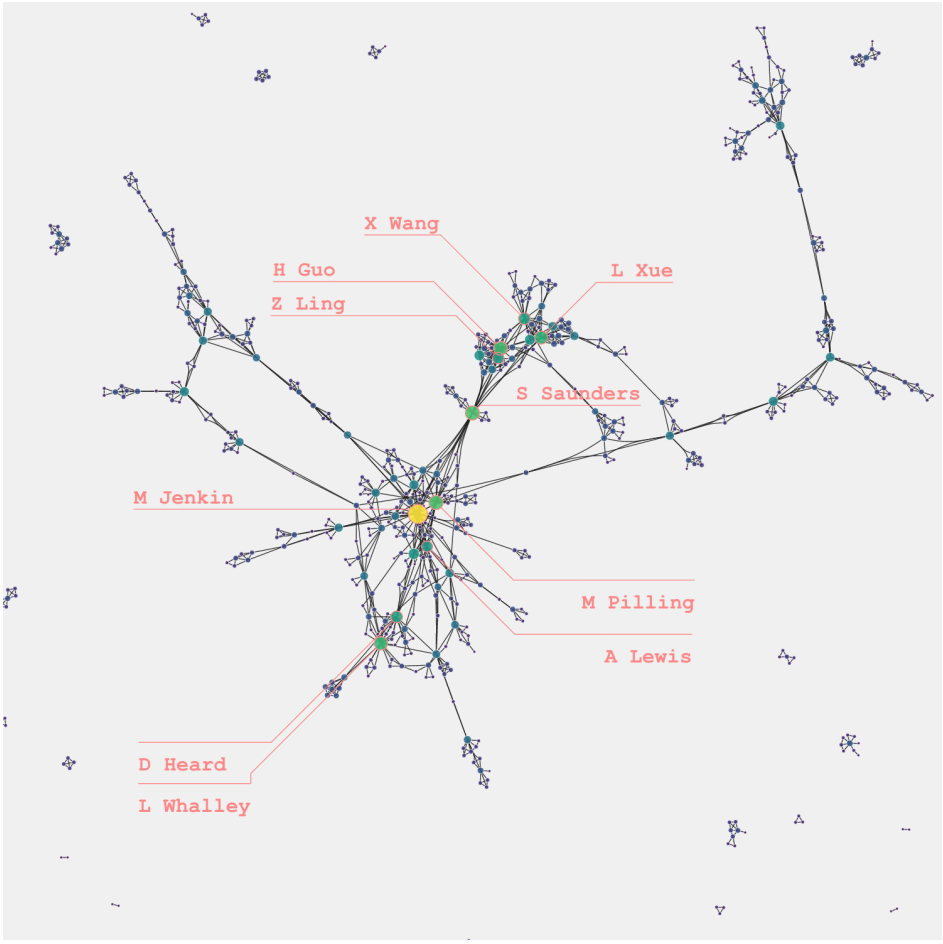
To demonstrate the information provided by different centrality metrics, a simple intuitive network (the co-author network in Figure 2.4) shall be used. This subsection will access the efficiency of graph centrality metrics in their ability to identify important nodes within a network.

2.0.11 Degree Centrality

The simplest, and most intuitive, metric is degree centrality [Freeman, 1978]. This is described as the sum of all links incident on a node - simply put, we count the number of edges going in and out of a node. This gives us an idea of the importance of a node and has been used to calculate influence within social media or the probability of a profile committing online auction fraud [Gemma, 2019; Freeman, 1978].

For the author network, Figure 2.5 we see that many of the names on the list are either contributors

to the MCM or have worked with them at Leeds. It is also seen that the authors with the most collaborations, or links, are very likely to appear within the most cited or citing papers (Table 2.2 and Table 2.3 discussed below). This is likely because both development (well-cited) and the evaluation/usage (well citing) of a mechanism requires knowledge from a range of different fields, making it an interactively collaborative process.



M Jenkin	39
S Saunders	25
M Pilling	25
H Guo	24
L Whalley	23
L Xue	22
D Heard	19
X Wang	19
Z Ling	18
A Lewis	17

Figure 2.5: **Degree Centrality.** In applying the degree centrality to the co-authorship network, it is possible to pick the authors with the greatest number of papers, of which the top 10 have been listed.

Directed Degree

For graphs where link direction holds an inherent meaning regarding their representation (for example in the citation graph an outward link symbolises that paper citing the one that the link points to), it is possible to further divide the degree centrality metric into inwards and outward links. This can allow us to separate items which provide a large number of lots of information (in-degree) and those who collate or collect it (out-degree). In applying these metrics to the directed citation graph, it is possible to get an insight into the core MCM development papers (Table 2.2) and separtate them from those which make use of the mechanism as part of a greater study (Table 2.3).

Protocol for the development of the Master Chemical Mechanism, MCM v3 Part A tropospheric degradation of nonaromatic volatile organic compounds	Saunders et al. [2003]
Protocol for the development of the Master Chemical Mechanism, MCM v3 Part B tropospheric degradation of aromatic volatile organic compounds	Jenkin et al. [2003]
Development of a detailed chemical mechanism MCMv3. 1 for the atmospheric oxidation of aromatic hydrocarbons	Bloss et al. [2005]

Table 2.2: **In-Degree of the citation network**: The top 3 most cited papers.

The MCM v3.3.1 degradation scheme for isoprene	Jenkin et al. [2015]
Atmospheric photochemical reactivity and ozone production at two sites in Hong Kong Application of a master chemical mechanismphotochemical box model	Ling et al. [2014]
HOx budgets during HOxComp A case study of HOx chemistry under NOxlimited conditions	Elshorbany et al. [2012]

Table 2.3: **Out-Degree of the citation network**: The top 3 most citing papers.

2.0.12 Closness Centrality

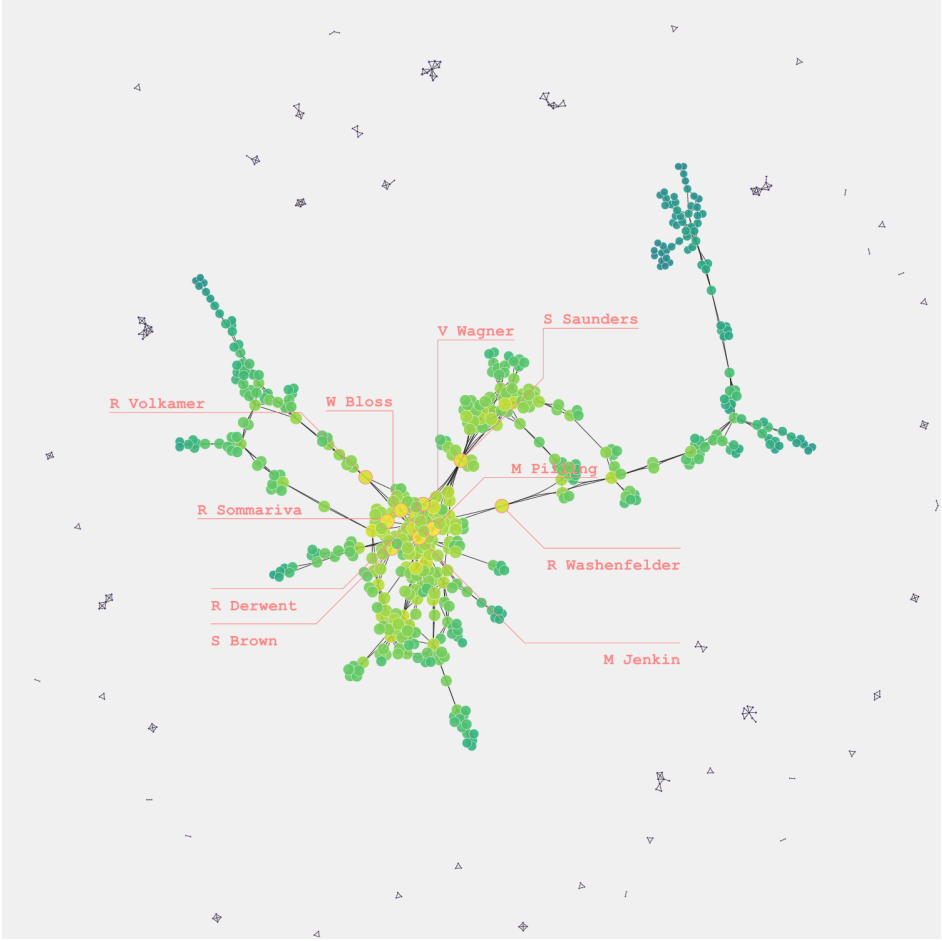
Often within a network, we are interested in how easy it is to to get information from one node to every other node. This is what the closeness centrality tells us. To calculate a nodes closeness we begin by taking the reciprocal sum of all the Dijkstra paths[5] to every other node [poliaktiv, 2011; Sabidussi,

[5]The shortest available path.

1966]. This gives is a representation of how far information from a certain will need to travel to reach every other node. Such a metric has applications in intelligence gathering, telecommunications and word importance within key-phrase extraction [Krebs, 2002; Borgatti, 2005; Boudin, 2013].

> ***Example analogy:*** *If we take the UK rail network as an example, York station will have a high closeness value as it is well connected and central in location. This means it is easy to reach every other location when compared to other stations.*

For the co-authorship network, Figure 2.6, nodes have been coloured by their closeness value. Here a heat-map-like effect may be observed, showing that information between the dense Leeds-York cluster is easier to disseminate across all parts of the graph than that of localised branches of authors less involved with the development team. The results of the closeness centrality suggest that should a problem (bug) or improvement (update) occur, Michael Pilling would be the best served to pass that information to all other groups using the MCM.



M Pilling	0.149995
M Jenkin	0.146532
R Sommariva	0.145251
W Bloss	0.144052
S Brown	0.142059
S Saunders	0.140176
V Wagner	0.139281
R Derwent	0.136450
R Volkamer	0.136184
R Washenfelder	0.135918

Figure 2.6: **Closeness centrality within the co-Author network.** Here a colour/size gradient is seen, with the nodes that are more central (in location) and better connected having a higher closeness than those in the peripheries - which are harder to get to.

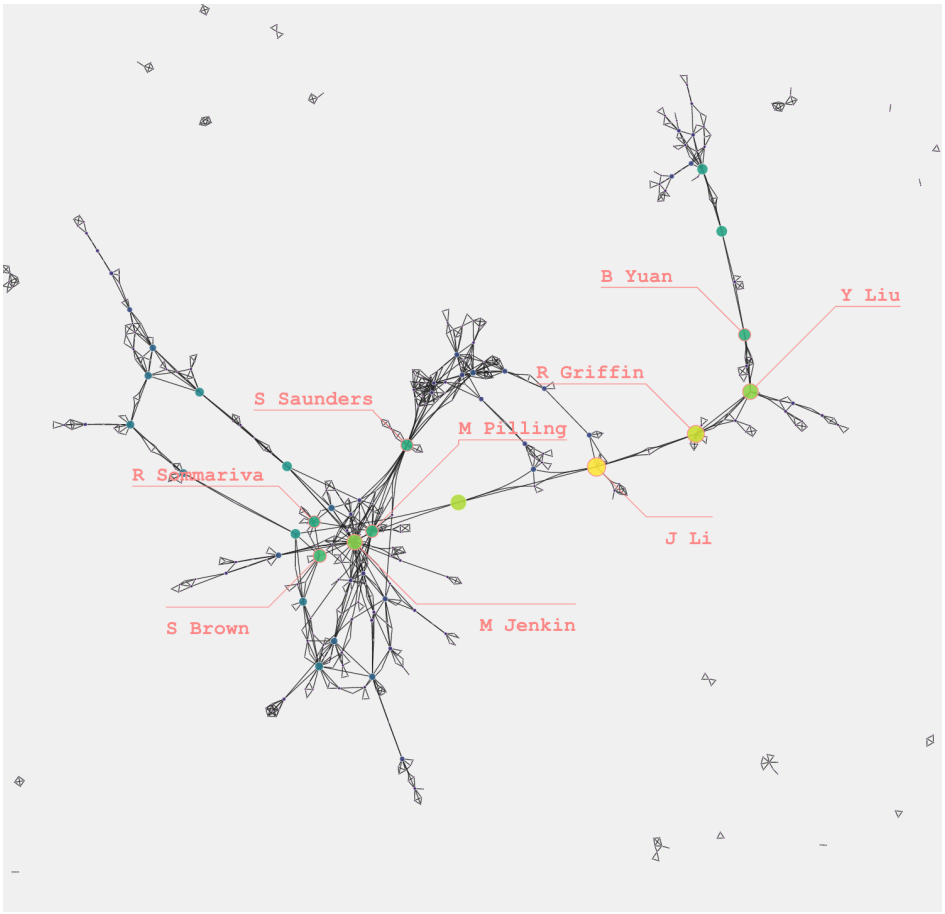
2.0.13 Betweenness

In social networks, it is often important not only to know who has the greatest reach (closeness centrality) but also where bottlenecks or 'broker' positions occur. Nodes with a high betweenness control, or limit, the amount of information that can be transferred across the network. If a node lies on a geodesic (the shortest path between two other nodes), we may consider it a 'pivotal' node, due to its role within the network [Needham and Hodler, 2019]. Should such a node then be removed, the

overall flow of information incurs either a deviation, the information will either need to travel a longer (alternative) route or may not be able to reach its destination at all [Freeman et al., 1991; Freeman, 1977; Brandes, 2001; Borgatti, 2005]. Betweenness centrality is a count of the number of geodesics which pass through a node. If multiple 'shortest' paths are possible, this is accounted for within the denominator.

> ***Example analogy:*** *Expanding on the UK rail network analogy, Shrewsbury station serves the critical role of connecting many lines from England to Wales. In removing this station, routes from the Liverpool or Manchester to Cardiff will be greatly increased. Additionally, the Aberystwyth section of the line will then become isolated from the rest of the country.*

Authors with a high betweenness in Figure 2.7 are seen to lie along the joints between clusters. Here we can imagine that removing Li, Griffin or Liu can disrupt the overall flow of collaboration, potentially isolating the work of the Max Planck from that of everyone else. Similarly, Jenkin and Pilling can be seen as holding much of the Leeds cluster together. In removing them from the network (if for example, the refused to collaborate) it is possible to see how many of groups within the Leeds environment may not have worked together, with the cluster potentially separating into several smaller groups. Finally, we see Saunders (Australia), who served to introduce the MCM to the Chinese atmospheric community. In removing her from the network, it can be seen that much of the collaboration which exists would have been significantly less likely.



J Li	0.180998
R Griffin	0.162558
R Washenfelder	0.153024
Y Liu	0.142194
M Jenkin	0.139818
S Brown	0.110188
M Pilling	0.102816
B Yuan	0.099914
S Saunders	0.097255
R Sommariva	0.094757

Figure 2.7: **Betweenness centrality within the co-Author network.** Nodes which lie on a pivotal position (connecting/bottleneck) tend to have a high betweenness value due to their crutial role within the network.

2.0.14 Spectral Methods And Matrix Analysis

Graphs can often be represented in the form of relationship (adjacency) matrixes (ref Chapter 1). This allows us to apply the theory of linear maps, such as eigenvectors and values, to stochiometric data in matrix form. Such methods have been around since the 1950s, [R. Seeley, 1949], but mainly became popular with the release of Larry Page's page-rank algorithm [Page et al., 1999] - the algorithm that began google. These methods, in addition to the HITS algorithm Table 2.0.14, make use of a graphs

native matrix representation to calculate node importance. Spectral algorithms can be broken down into four categories [Vigna, 2016]:

	No Normalisation	Row Normalisation
No Damping	Eigenvector [Bonacich, 1987, 2007]	Markov Chain Steady State [R. Seeley, 1949]
Damping	Katz [Goh et al., 2001]	Total Effect Centrality PageRank [Page et al., 1999]

Here damping terms represent the probability of moving to the new random starting position, allowing for the user to 'randomly select a new webpage' or leave an isolated cluster. The normalisation of the matrix does not affect the node ranking, but merely adjusts the numerical output of the algorithm. It is for this reason that its overall practicality may be debated [Vigna, 2016]. Since page rank is the most common of these methods and allows for a tune-able degree of randomness within network propagation. This is discussed in more detail in the next subsection.

Hypertext Induced Topic Search (HITS)

A common eigenvector algorithm used for classifying webpages is the HITS algorithm. This helps categorise the role of a node as either a Hub or an Authority, [Kleinberg, 1999; Langville and Meyer, 2005; Kumar and Upfal, 2000]. Similar to the in and out-degree metrics, this algorithm separates nodes with many outgoing links (an authority) from those with many ingoing ones (an information hub). Overall this provides similar results to the in/out degree, although since it looks more on how information propagates across the network as a whole, it often provides more accurate, and different, rankings to simple degree analysis.

2.0.15 Page Rank

Arguably the best-known centrality algorithm is PageRank. This is a spectral method for measuring the transitive influence of a node, by taking the effect of neighbours and by their neighbours into account [Needham and Hodler, 2019]. The page rank algorithm was initially developed to provide a better way of ranking web pages [Page et al., 1999]- here an important page is not only one of many links, but links to other important sources. In the context of academic papers, that same paper also found that in predicting future citations, the page rank algorithm fared better than using the current citation count of a paper. To explain how this works, we will look at the mathematics behind the algorithm, and then eventually apply it to the co-authorship graph in Subsubsection 2.0.15.3

2.0.15.1 The Google Matrix

To solve for page rank, a google matrix must first be constructed. Once done this is iterated until convergence is reached.

To build a google matrix, we must first generate a dyadic link map of the graph[6] - its adjacency matrix $A_{i,j}$ (i, j are the source target indexes). This is then converted into a Markov matrix $M_{i.j}$ by dividing each column j by the sum of the total outgoing links of node j, Algorithm 1. Species with no outgoing links (sinks), are adjusted with either a personalised list of values or the constant $1/n$, (where n is the number of nodes) to replace the zero-sum columns. This produces a normalised[7] matrix of Markov chains representing the fractional production for node j from all other nodes.

Algorithm 1 Adjacency to Markov matrix.

1: Obtain graph adjacency matrix, $A_{i,j}$.
2: **repeat**
3: **for each** $j \in$ columns **do**
4: $M(:, j) \leftarrow A(:, j)/\Sigma_{i=1,n}A(j, i)$
5: **end for**
6: **until** $\Sigma_{i=1,n}M(i, j) = 1$

The google matrix $G_{i,j}$ can now be defined using Equation 2.1. Cyclic reactions and nodes that only point towards each other within a group can 'trap' the user, increasing their ranks. To account for this, a damping factor, typically $\beta = 0.85$, is used. This defines the probability that the user follows a link, and that for which they randomly select another page: $(1 - \beta)$ [8]. The damping factor used varies greatly with the application, with values such as $\beta = 0.694$ having been found optimal for the use of biological data [Hobson et al., 2018].

$$G_{i,j} = \beta M + \frac{1 - \beta}{n} \tag{2.1}$$

β - *Probability the user follows a link*

$(1 - \beta)$ - *Probability the user does not follow a link (teleportation)*

n - *Number of items / species*

M - *Normalised markov matrix*

[6] In sociology a dyad is a group of two people - the smallest possible social group.

[7] $\Sigma_{i=1,n}M(i, j) =$ unity

[8] Also known as teleportation.

2.0.15.2 Solving The Algebra

Once defined, the google matrix is solved by propagating a one's vector, r of length n, where n is the number of species using Algorithm 2.

Algorithm 2 Solving the google matrix linear algebra

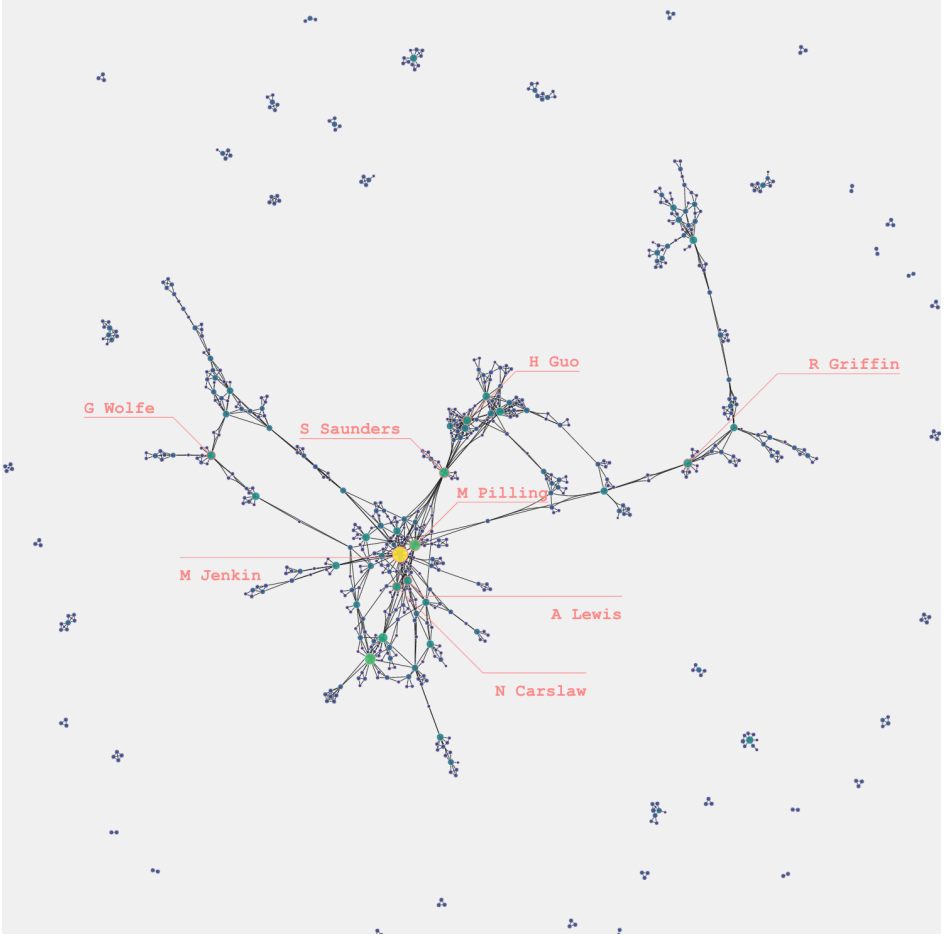
1: Define value vectors $\bar{r}_t$ and $\bar{r}_{t+1}$:
2: $\bar{r}_t = [1_1, 1_2, ..., 1_n]$, $\bar{r}_{t+1} = [0_1, 0_2, ..., 0_n]$
3:
4: **while** $||\bar{r}_{t+1} - \bar{r}_t|| > \epsilon$ **do**
5: $\quad \bar{r}_{t+1} \leftarrow M.\bar{r}_t$
6: $\quad \bar{r}_t = \bar{r}_{t+1}$
7: **end while**

This is repeated until a pre-defined tolerance, ϵ is reached. For best results, this can be set to just under the numerical precision of the programming language/hardware.

For smaller systems, it is possible to use the LAPACK [lap, 2019] library, as used by [Oliphant, 2006]. For a large network, however, the computation of an $n \times n$ matrix can be very memory inefficient for small machines. It is then possible to apply the methods as described above using a sparse matrix on per-node bases as can be seen within the scipy implementation of the networkx source code [Jones et al., 01 ; Hagberg et al., 2008].

2.0.15.3 Prediction

As the PageRank algorithm is a physical representation looking at how quantities 'flow' within a network, it can be used to identify not only the bottlenecks (betweenness centrality) but also any nodes which are connected well within the network. As the flows between a node are somewhat governed by the number of links it contains, the PageRank algorithms tend to correlate, but not a dependance, on the betweenness of a node. Figure 2.8 shows the PageRank algorithm to identify important authors within each 'cluster' or research group. Due to its propagating nature authors connected to these important nodes are often also of greater importance. An application of this can again be the determination of how to best spread new results or information with the least number of people. *Note: if we only had one person we would probably use the node with the highest closeness centrality.*



M Jenkin	0.010435
L Whalley	0.006589
M Pilling	0.006488
S Saunders	0.005591
D Heard	0.005192
N Carslaw	0.004833
H Guo	0.004594
G Wolfe	0.004523
A Lewis	0.004508
R Griffin	0.004500

Figure 2.8: **Page Rank centrality within the co-Author network**.

2.0.16 Conclusions

In this section, we have explored the use of centrality metrics to provide us with information on an unweighted co-authorship network of the MCM. Having used these to demonstrate the different roles that may be extracted from a node, we can move on to applying them to a chemical mechanism. In the next section, a global set of metrics will be used to determine the network type/structure of the MCM. Once this has been done, graph construction using simulation results (a weighted graph) will be looked into in Subsection 2.0.23.

2.0.17 Classifying the Master Chemical Mechanism network

Having shown that graph metrics can help the roles of individual nodes within the network, I will now apply them to a chemical system. Since computational efficiency and resources are often a limiting factor, many applications of the MCM only require a small subset of the entire mechanism. For this reason, it may be of interest to compare these against each other, in an attempt to classify the type of network the MCM chemistry falls under. In this section, we apply graph theory to the entire MCM network, to determine its defining characteristics. This is achieved through the analysis of several hundred randomly selected subsets of the MCM.

2.0.18 Network Density

Network density is the easiest to understand. Visually this can induce complexity and obscure aspects in a graph, mathematically it can greatly increase the computation time for metrics or algorithms. By definition, we can define network density as a measure of how well connected a node is to every other node, mathematically it is the ratio of edges against the total number of possible edges for a complete graph[9] of the same size. In chemical terms, we can use this to determine the sparsity of the graph (which has applications on model integrator selection) and give us insights on the chemical structure. In Figure 2.9 the addition of more species (nodes) results in an overall decrease in the node-edge ratio - it's density. This suggests a modular or hierarchical structure, where new species directly react only with a set number of species, and not the entire mechanism. An explanation for this is that the addition of larger species introduce new branches within the chemistry, which then need to be oxidised before they are small enough to react with the species from a different branch. Since these branches are somewhat isolated from the rest of the chemistry, they decrease the network density, even though their addition may increase the amount of chemistry that occurs within it.

[9] A complete graph is one where every node is connected to every other node.

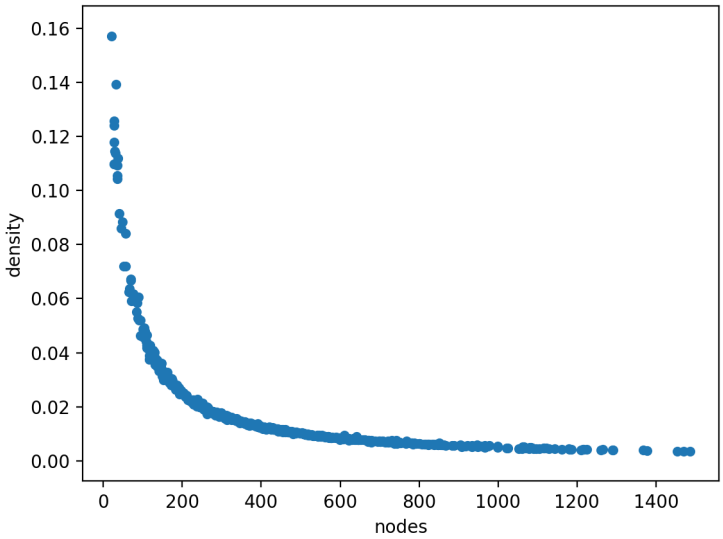


Figure 2.9: **How the MCM graph density scales with number of species.** A figure showing that an creasing the number of species within a mechanism subset results in an increased model sparsity (decreasing density).

2.0.19 Small World Phenomena

Within the biological or social sciences the small world phenomenon, colloquially known as 'six degrees of separation', is a common occurrence within network structure [Watts and Strogatz, 1998]. Such networks have a large number of localised clusters (cliques) all with a short path length between their elements [Humphries and Gurney, 2008]. This makes it easy to reach all parts of a network with only a couple of hops/reactions. In the initial interactive explorations of graph visualisation, it was found that in selecting the reactions of a node, and consequently the reactions of all the nodes which react with them, very quickly a large proportion of the network was highlighted. This suggests that the network may follow the small world phenomena, especially as it is a sparse network, Subsection 2.0.18.

One of the possible methods for establishing the small world-ness of a graph falls under the of the omega (ω) coefficient:

$$\omega = L_r/L - C/C_l \tag{2.2}$$

Here C is the average clustering coefficient and L, the shortest path length of the graph. Comparing these with the average shortest path length, L_R, and clustering coefficient C_l (as calculated using an equivalent random and lattice graph) gives the above equation. The output is a result between positive and negative one {-1,1}, where a value of 0 suggests the graph exhibits perfect small world-ness.

In assessing the network structure of the MCM, a Monte Carlo (random) approach was taken to

extract several hundred subsets from the entire mechanism. For each of these, the omega coefficient was calculated and plotted in Figure 2.10. Here it is seen that subsets with a small number of species (for example those derived only from Methane or Ethane) exhibit a more lattice-style graph, with the majority of the networks showing a more random network structure. All the results, however, show a prevalence of small-world features over any of the alternative network structures - they are closer to 0 than 1 or -1. This reflects the idea that large species react locally, forming branches (REF VIS CHAPTER), before oxidising to smaller species with more reactions. This result is also seen within the Reaxys chemical database [Jacob and Lapkin, 2018].

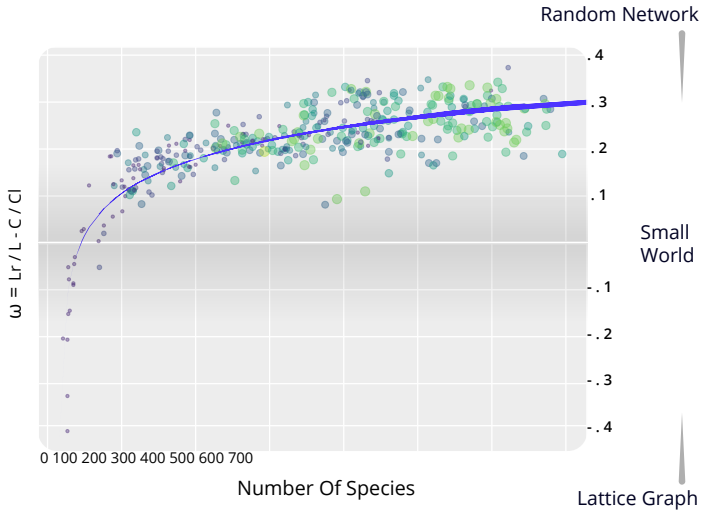


Figure 2.10: **A figure showing the small worldness for many Monte-Carlo selected MCM subsets**. The network structure of these is then assessed using the omega coefficient, with [-1,0,1] corresponding to the perfect lattice, small-world and random network structure. Here Node size and colour represents the number of reactions in the subset and the number of primary VOCs (blue=small, green=large).

2.0.20 Power Law And Scale-Free Graphs

In real-world applications, it is common to have a hierarchical structure. These are often seen in the increase of citation counts in academic papers [de Solla Price, 1965], email threads [Ebel et al., 2002] and the world wide web [Needham and Hodler, 2019]. Unlike random or small-world graphs, scale-free graphs take a hub-and-spoke structure (Figure 2.11), which follows a power-law distribution - that is that scaling probability $p(x) \propto x^{-\alpha}$, where α is a constant and known as the scaling parameter.

Broido and Clauset [2019] suggests that scale-free networks are rare, and often misdiagnosed with incorrect tests, or the misinterpretation of power-law features in a network. Similarly, Clauset et al. [2009] suggests that even if the data distribution of a graph is well represented by the power-law distribution, in many cases a logarithmic or exponential distribution may have a better fit.

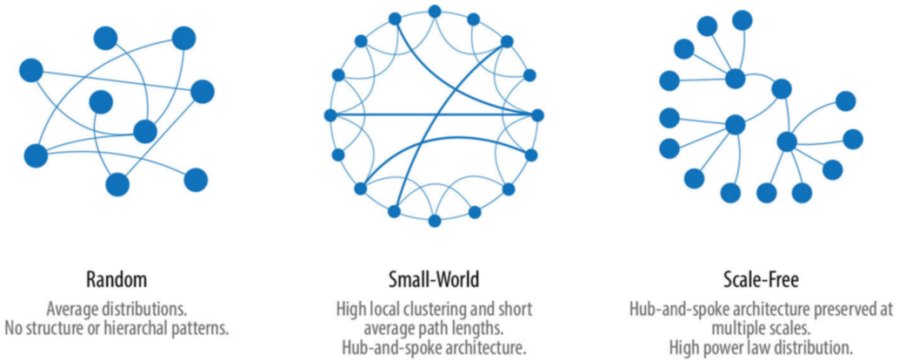


Figure 2.11: **The different network structures**. A visual depiction of the different graph structures. Source: Needham and Hodler [2019]

To assess the best distribution for describing the monte carlo subsets of the MCM I use the Kolomogorov-Smirnov statistic [Press et al., 1992]. This calculates the maximum distance D between the selected cumelative distribution function $S(x)$ (In our case the Logarithmic, Exponential and Power Law) of the data and the fitted model $P(x)$:

$$D = \max_{x \geq x_{min}} |S(x) - P(x)| \tag{2.3}$$

Using the MCM subsets from before Figure 2.12 shows that out of the three tested distributions, the MCM is best represented as a power-law distribution. Although this is not entirely within the chosen 5% significance, it is highly indicative that some aspects of the network are scale-free.

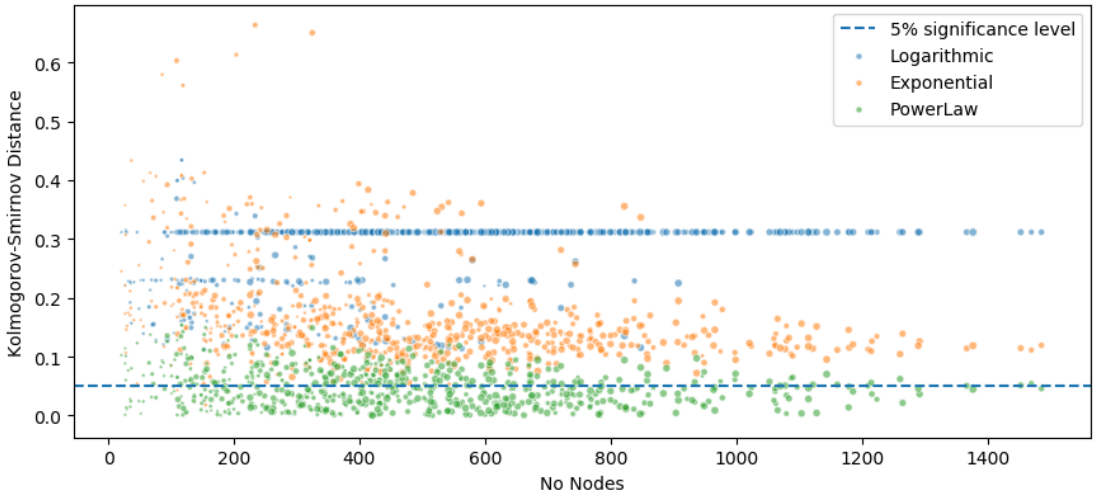


Figure 2.12: **Comparing the MCM subsets against a power law, logarithmic and exponential distribution.** The fit for different cumulative probability distributions of nodes in the MCM network is compared to determine the type of network hierarchy the chemistry follow. This is done by comparing the distance of the calculated distribution of data against a perfect one using the Kolomogorov-Smirnov test. The closer the two distributions are the better the fit.

2.0.21 Describing The MCM Network

To conclude the MCM network exhibits both small world and scale-free (power-law) characteristics. This agrees with previous knowledge about the apparent network structure (branch and core - ref CH1/2). Here large primary emitted hydrocarbons produce branches of a hierarchical nature, as they are progressively broken down into smaller species. Since smaller species are then able to react with a much greater range of species, they then begin to form a tightly connected core, which exhibits many small-world features. This can be seen as the densely connected region within the graphs in CHAPTER !.

Having classified the MCM network type, the next section will look at how MCM based simulation results can be converted into the graph structure for more in-depth analysis, Subsection 2.0.25.

2.0.22 Graph Construction methodology

Thus far we have only applied a qualitative analysis on the relationships between species in a mechanism. Although this can educate us about the chemistry within a specific system, often a quantitative value for the rate of reaction between different species is required when undergoing scientific evaluation or policy advice. To obtain such results a chemical mechanism is placed within an atmospheric model, initial concentrations are supplied and the chemistry is propagated forwards[10] in time. Currently, there exist three main model diagnostics which we may use to analyse the importance or role of a species from a simulation (model) output.

2.0.22.1 Concentration Time Series

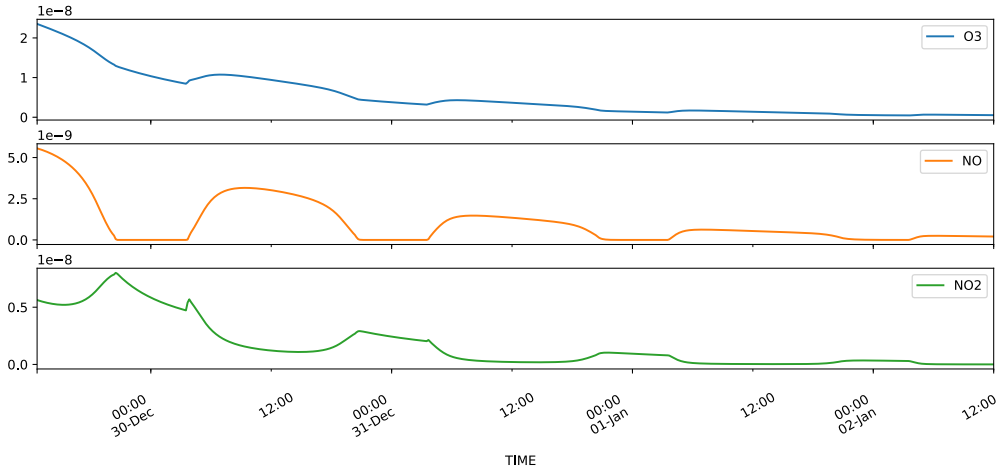
The simplest of these methods look at the abundance of a species at a specific point in the atmosphere - its concentration. As time moves forwards, chemicals within the atmosphere undergo a range of reactions which result in the making and breaking of bonds - thus the changing from one species to another.

Using the species concentration as a metric, we can map how it changes over time, and how in changing the initial concentrations of a simulation can produce different results. This can be useful for looking at a range of possible scenarios and evaluating the potential outcome after a pre-determined amount of time. An example would be through the use of policy-based simulations to predict changes in ...

Using a simple example from a Methane only subset of the MCM (Figure 2.13), it is possible to observe the inverse relationship between NO_2 and NO using only their concentration profiles. Here nitrogen

[10] Or backwards if the adjoint is used. (see section PAGERANK APPLICATIONS)

monoxide reacts with a RO_2 species to produce an RO and nitrogen dioxide. This then photolyses back to nitrogen oxide, releasing oxygen which may go on to form ozone (REF NOX CYCLE IN INTRO). The latter part of this reaction is dependant on photons and therefore can only occur during daytime (mostly).



$$\mathrm{NO} \underset{h\nu}{\overset{\mathrm{RO_2}}{\rightleftharpoons}} \mathrm{NO_2}.$$

Figure 2.13: **A concentration time series from a simple methane-only simulation.** This is the simplest method for identifying changes in species within a model simulation. This multi-plot shows the changes in concentration profiles for all initialised species (NOx:10ppb; CH_4:20ppb; O_3:30ppb) following an initial 3 day spin-up to steady state.

2.0.22.2 Rate Of Production And Loss

Analysing the concentration-time profiles allows the comparison of how a series of scenarios or runs change concerning their initial conditions and simulation length. Although these can tell us how, and how much, each species changes over time it does not rank or quantifies the specific reactions to which this may be attributed. Rate of Production Analysis (ROPA)[11] provides a method for establishing the total contribution from each reaction by calculating the change of concentration (concerning time) for the produced species - the instantaneous reaction Flux.

$$r_1 = \eta A + \omega B \xrightarrow{\kappa_1} C \qquad \text{Reaction 1} \tag{2.4}$$

$$f(C) = \frac{\delta C}{\delta t} = [A][B]\ \eta\omega \times \kappa_1 \qquad \text{Instantanious Flux } (\Gamma) \tag{2.5}$$

Here A, B and C are example species; [A],[B] and [C] are species concentrations; η and ω are rate coefficients and κ is the rate of the reaction.

[11] and loss

Using a sample simulation representative of the conditions within Beijing (an urban environment), we explore the reactions contributing to the production and loss of CH_3CO_3, Figure 2.14 at noon. The main reason for this specific example is that it can demonstrate how isolating a specific cause for the change within a species concentration may prove difficult in the context of atmospheric chemistry. Here we have many similarly weighted production and loss reaction, including that of peroxyacetyl nitrate (PAN) and nitrogen dioxide: $CH_3CO_3 + NO_2 \rightleftarrows CH_3C(O)ONO_2$ (PAN). The reversible nature, coupled with its near-identical production and loss fluxes produce a very small net change within our species of interest (CH_3CO_3). Although this may be seen by calculating the cumulative flux between individual species, it is evident that simply looking at the concentrations or highest-ranking reaction fluxes may not be the best method of determining influence. To account for this we can look at how a change in one species can affect another using the Jacobian method.

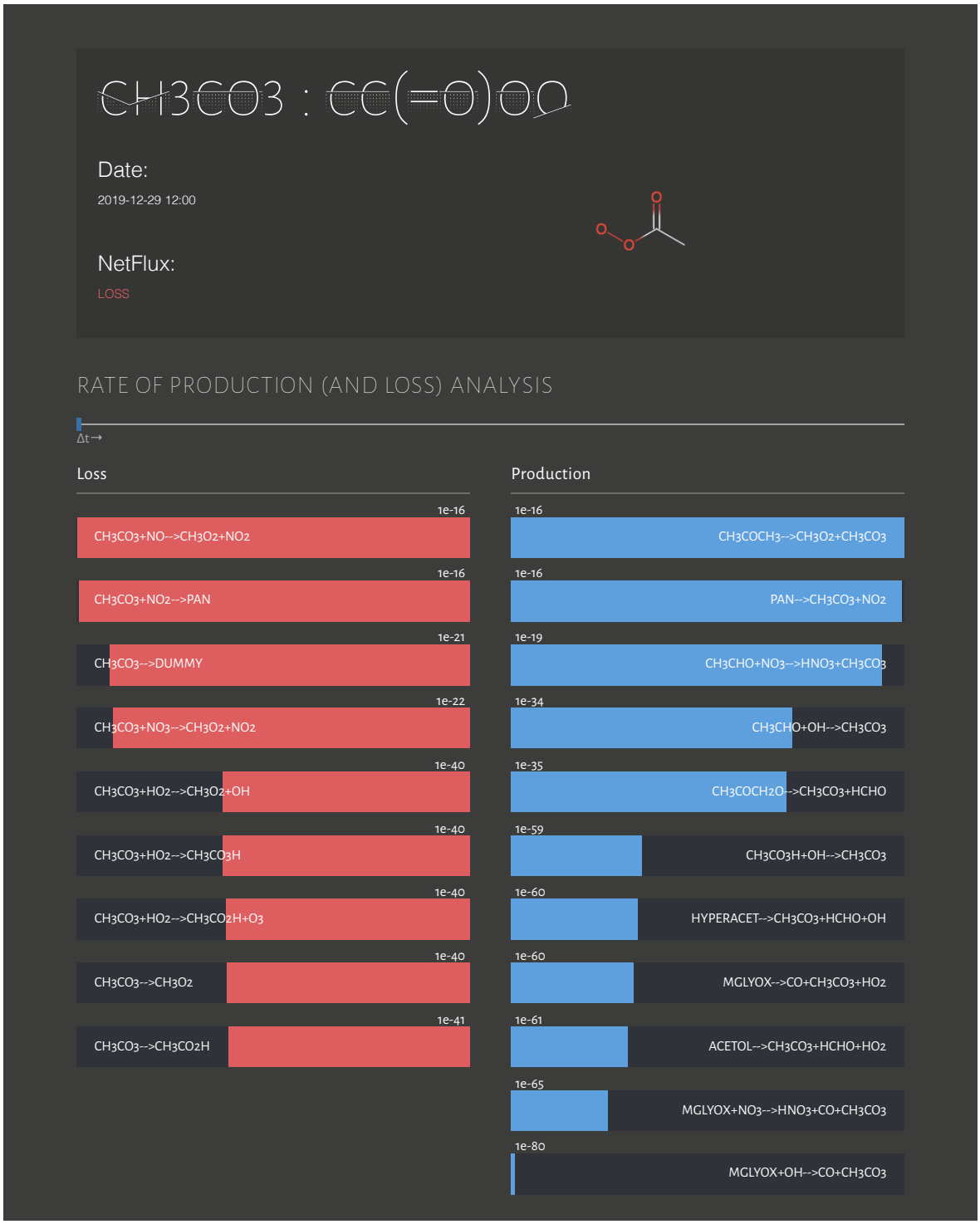


Figure 2.14: **Rate of production and loss analysis plot for CH_3CO_3 exhibiting a net loss (daytime).** An example ROPA plot from a simulation representing the chemistry within Beijing. This is used to identify the usefulness and weaknesses of using such a method.

2.0.22.3 The Jacobian

"The Jacobian [matrix] generalises the notion of gradient to describe the sensitivity to a vector" - Brasseur and Jacob [2017]. That this means is that in taking the partial derivatives of each reaction flux (e.g. from Equation 2.5), we can construct a representation of the influence each species has on itself - for example, the influence of species A on C and B on C (Equation 2.6-2.7).

$$\frac{\partial}{\partial A} \cdot \frac{\partial C_{r_1}}{\partial t} = \eta\omega B\kappa_1 \qquad \Gamma \text{ influence from A} \tag{2.6}$$

$$\frac{\partial}{\partial B} \cdot \frac{\partial C_{r_1}}{\partial t} = \eta\omega A\kappa_1 \qquad \Gamma \text{ influence from B} \tag{2.7}$$

These partial equations can then be aggregated for all reactions that contain the two species - taking the effect of species B on species C, for example, produces Equation 2.8. Using these aggregate sums it is now possible to construct a pairwise relational matrix describing the influence each species has on every other species- Equation 2.9. This is known as the jacobian matrix and is what is used to propagate the chemistry within a simulation forwards in time.

$$\mathbf{J}_{C,B} = \frac{\partial f(C)}{\partial B} = \frac{\partial}{\partial B} \cdot \left(\frac{\partial \Sigma_{r_1}}{\partial t} + \frac{\partial \Sigma_{r_2}}{\partial t} + \cdots + \frac{\partial \Sigma_{r_n}}{\partial t} \right) \tag{2.8}$$

$$\mathbf{J}_{i,j} = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} & \cdots & \frac{\partial f_1}{\partial v_n} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} & \cdots & \frac{\partial f_2}{\partial v_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial v_1} & \frac{\partial f_n}{\partial v_2} & \cdots & \frac{\partial f_n}{\partial v_n} \end{bmatrix}^{n,n}_{i,j=1} \tag{2.9}$$

2.0.23 Graph Construction Methodology For Simulated Data

Having covered the general definition of a Jacobian matrix and how it is constructed, we can now apply it to the context of mechanism analysis and comprehension. The first analogy that needs to be made is that for the flux, we have a the first differential of a specific reaction in time. If we consider the change in a species concentration as a 'displacement', we can think of the flux as its 'velocity'. Similarly, the Jacobian provides us with a description of how the individual flux of a species changes concerning the concentration (or displacement) or another species (the second-order partial differential). This is analogous to the acceleration of the object or particle we first displaced. In using the jacobian, we have constructed a relational matrix which outlines the effect a 1% change of a species has on all other species - a concept which is the foundation of the connectivity method (a mechanism reduction technique where all but essential and important species are removed), [Turányi and Tomlin, 2014].

Since the format of a jacobian is already in the form of a relational matrix, it can easily be converted to a weighted adjacency matrix, and then directly into the graph format. Since it only considers the aggregated influence between species, much of the work that would otherwise be needed to convert a mechanism into a graph format has already been done. To make use of the Jacobian matrix, several extraction algorithms were written for an updated version of the Dynamically Simple Model of Atmospheric Chemical Complexity (DSMACC) [DANDSMACC,DSMACC ref], as discussed in INTRODUCTION. Here we edit the kinetic pre-processor output, [**?**] to release the values of the Jacobian Matrix and return them at each model timestep for analysis. The process for how this is done is described in Subsection 2.0.24.

A Note On Using The Flux Instead Of The Jacobian

Depending on the model setup or the users' capabilities, extraction of the jacobian matrix for each timestep may not be possible. In many cases, the reaction rates and concentration may still be available, allowing for the calculation of reaction fluxes throughout the simulation. If this is the case the total flux can be calculated using the method described in Equation 2.5. From this, an edge-weighted by a reaction flux can be created from every reactant to each product. This generates a multi-graph[12] which may be simplified by taking the net flux value for all edges between two nodes.

However, the potential for human/coding error, additional simplification and a non-explicit definition of the contribution of each species make the use of a Jacobian much more efficient in network generation from a chemical mechanism.

[12] A graph with multiple edges between nodes

2.0.24 A Practical Example Using The MCM

Taking a single equation from the MCM we may calculate the jacobian relationships between species and convert them into a graph. A randomly chosen ethane reaction (Equation 2.10) from a simple mechanism was chosen. It must be noted that in general it is unusual in the MCM that alkyl radicals react rapidly and extremely well with O_2 to from stabilised peroxy radicals, [Jenkin et al., 1997]. In general, the reaction would consist of the following two steps: $C_2H_6 + OH \xrightarrow{\kappa_1} C_2H_5\cdot + H_2O$ and $C_2H_5\cdot + O_2 \longrightarrow [\kappa_2]\, CH_2H_5O_2$.

$$C_2H_6 + OH \xrightarrow{\kappa_3} C_2H_5O_2 \tag{2.10}$$

For simplicity in this example, this will be the only equation for our mechanism. The resultant Flux Equation 2.11 and resultant Jacobian Equation 2.12 may be calculated.

$$\Gamma = [C_2H_6][OH]\kappa_1 \tag{2.11}$$

$$\mathbf{J}_{i,j} = \begin{bmatrix} \frac{\partial f_{[C_2H_6]}}{\partial t\ \partial[C_2H_6]} & \frac{\partial f_{[C_2H_6]}}{\partial[OH]} & \frac{\partial f_{[C_2H_6]}}{t\partial[C_2H_5O_2]} \\ \frac{\partial f_{[OH]}}{\partial t\ \partial[C_2H_6]} & \frac{\partial f_{[OH]}}{\partial t\ \partial[OH]} & \frac{\partial f_{[OH]}}{\partial t\ \partial[C_2H_5O_2]} \\ \frac{\partial f_{[C_2H_5O_2]}}{\partial t\ \partial[C_2H_6]} & \frac{\partial f_{[C_2H_5O_2]}}{\partial t\ \partial[OH]} & \frac{\partial f_{[C_2H_5O_2]}}{\partial t\ \partial[C_2H_5O_2]} \end{bmatrix}_{i,j=1}^{3,3} \tag{2.12}$$

Since not all species react with all other species, we can remove reactions that do not exist. This forms a 'sparse' jacobian. Substituting numbers from a subset mechanisms containing the methane and ethane precursors, we get Equation 2.13.

$$\mathbf{J}_{i,j} = \begin{bmatrix} \frac{\partial f_{[C_2H_6]}}{\partial t\ \partial [C_2H_6]} & -2 \times 10^{-7} & 2 \times 10^{-7} \\ -0.1 & \frac{\partial f_{[OH]}}{\partial t\ \partial [OH]} & 0.1 \\ & & \frac{\partial f_{[C_2H_5O_2]}}{\partial t\ \partial [C_2H_5O_2]} \end{bmatrix}_{i,j=1}^{3,3} \tag{2.13}$$

This allows us to see two things. Firstly that with the absence of external intervention (e.g. emissions) the overall change of concentration is a conserved property. Secondly ...

Representing these relationships as a simple 'ball and link' style graph gives us Figure 2.15.

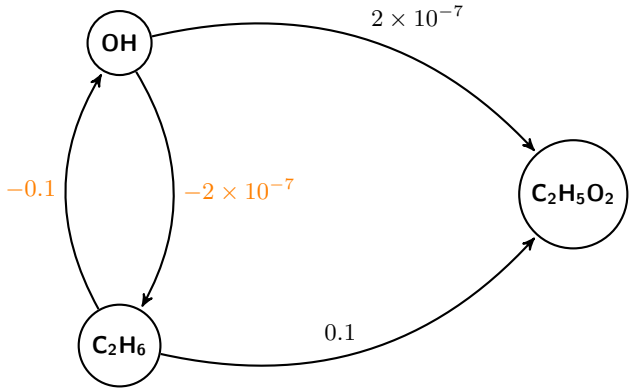


Figure 2.15: A graphical representation of Equation 2.13 derrived from the Equation 2.10

Converting the Jacobian into an adjacency matrix

Adjacency matrixes are a set of matrix representations which can be used in the construction of a graph. The relational data of the Jacobian matrix Equation 2.13 inherently holds such property and can be directly translated to produce a graph, Figure 2.15. However, we notice that some edge weights are negative, which although providing information about the chemical system provides no physical meaning in the graph structure.

It is for this reason that we can reverse the direction for all negative links to produce Figure 2.16.

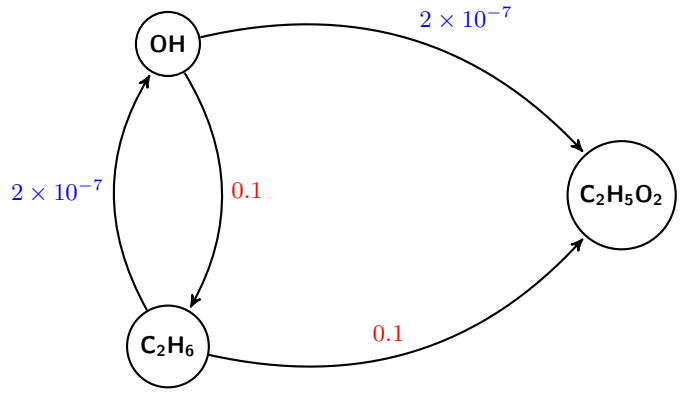


Figure 2.16: Reversing the directions on negatively weighted edges from Figure 2.15

For most graph algorithms this should be sufficient and is generally all that is needed. In some cases, it may, however, be noted that the graph may further be simplified to produce Figure 2.17. Although this is more practical, eigenvector metrics such as PageRank will automatically transfer the 'flow' of information down the system producing much the same overall result.

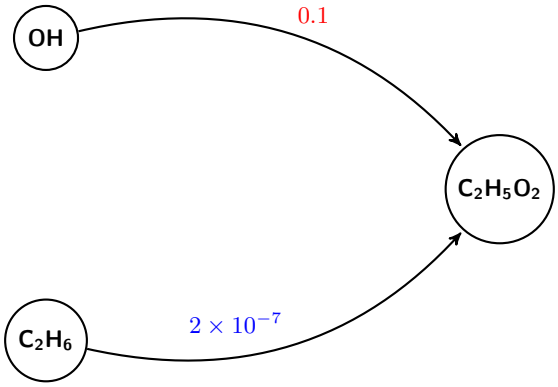


Figure 2.17: Simplifying Figure 2.16

2.0.25 Case study Example

In this section, the centrality metrics discussed in Subsection 2.0.10 are applied to a range of scenarios. These range from polluted urban environments such as London [REF] and Beijing [REF], to marine and terrestrial forest- Cape Verde REF and Borneo REF. We determine the main drivers for the chemistry and compare the species which are important across each simulation.

2.0.26 Establishing Initial Conditions From Observational Data

Within experimental data assimilation, it is not uncommon to face problems which result in unreliable or missing data. These can range from anything as little as measuring below the instrument sensitivity

to powercuts and equipment damage/theft from the local wildlife. This can result in problems when analysing the results and combining them to create a simulation of the chemistry for that environment.

To overcome this, traditionally a combination of data filtration, smoothing and interpolation are required. Although it is possible to fit a diurnal profile, through iterative methods of comparison, and cubic splines, a much simpler way would be to use a Multi-Layer Perceptron Regressor model (MLPRegressor) as provided by sklearn, [Pedregosa et al., 2011]. This is described below.

2.0.26.1 The Origin Of Artificial Neural Networks

The concept of a neural network originated within the field of neuroscience. In biological neurons, signals are sent through the use of electrical impulses using their synapses. When a sufficient number of signals are received within a short timeframe, a neurone will respond, often firing a range of its signals. Using this as a foundation,McCulloch and Pitts [1943] presented a computational model of the biological neuron - the artificial neuron. This has a series of binary inputs and produces a single binary output. This idea was later improved with the invention of the perceptron - a linear classifier which classifies categories by separating them with a straight line. Invented by Rosenblatt [1958], this was popularised as a device representative of a modern-day shallow neural network - [John Hay, 1960], Figure 2.18. Unlike the artificial neuron, however, the perceptron can take non-binary (numerical) inputs of an associated weight which allows for the computation of simple linear binary classification. Much like Logistic regression, the perceptron produces a positive or negative classification based on a certain threshold[13].

[13]It is worth noting that while a Logistic Regression classifier can output a class probability, the use of a hard threshold means that this is not done within the perceptron algorithm [Géron, 2017]

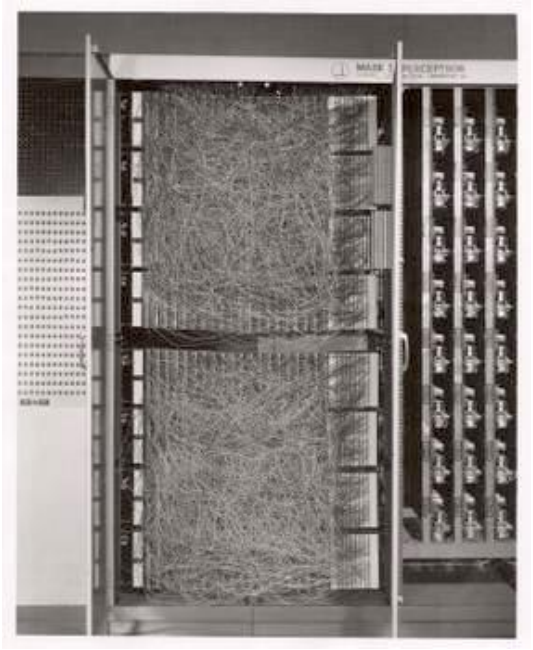


Figure 2.18: **The Mark 1 perceptron** Both software and hardware are different manifestations of a flow chart. The perceptron hardware accomplished what is now done using software. Source: Cornell [2020]

2.0.26.2 The Multi-Layer Perceptron

Limitations of the perceptron include the classification of complex patterns such as the XOR problem (where a category appears between two other categories e.g. 1|0|1 - this cannot be classified by a single linear split). In taking inspiration from nature, Figure 2.19, it is possible to overcome this with the use of multiple layers. This creates a deep ($>$ 2 two hidden (non-input) layers of perceptrons[14]) artificial neural network (ANN)

The multi-layer perceptron (MLP) model now represents a simple feed-forward network, much like a decision tree. However, unlike a decision tree, the MLP ANN can describe the probability a branch is taken using non-linear activation (threshold) functions. These are discussed in detail as part of **??**. The weighting thresholds for each neuron are then calculated by backwards propagation of results through the network until a suitably good result is produced.

> ***Example analogy:*** *Backpropagation can be likened to the iterative calibration of scientific instrumentation. In the field of atmospheric chemistry, laser-induced fluorescence is used to calculate species concentrations and reaction rates within the troposphere, [Dillon et al., 2006; Bloss et al., 2004]. Here the frequency of a laser can be adjusted in contrast with a known target (e.g. an amount of SO_2) to produce a response curve showing where the maximum resonance occurs.*

[14]These are sometimes referred to as Linear Threshold Units.

Similarly, a neural network can be 'trained' (calibrated). This is done through the use of a 'training dataset' - a set of input-output pairings which represent a random selection of 2/3rds of the total dataset. Next, the neurons within each layer (similar to the potentiometer dials on an instrument) are adjusted in sequence through the layers to match the known result (a standard of known concentration) to the input values provided. This process is repeated until for many iterations, or until a sufficiently 'good' prediction is attained for the entire training dataset (early termination). The power of ANNs comes from the ability to adjust neuron thresholds whilst moving both forwards and backwards through the network (Note: predictions of an MLP are still only passed forwards). Finally, model performance is evaluated against the remaining 1/3rd of the total dataset.

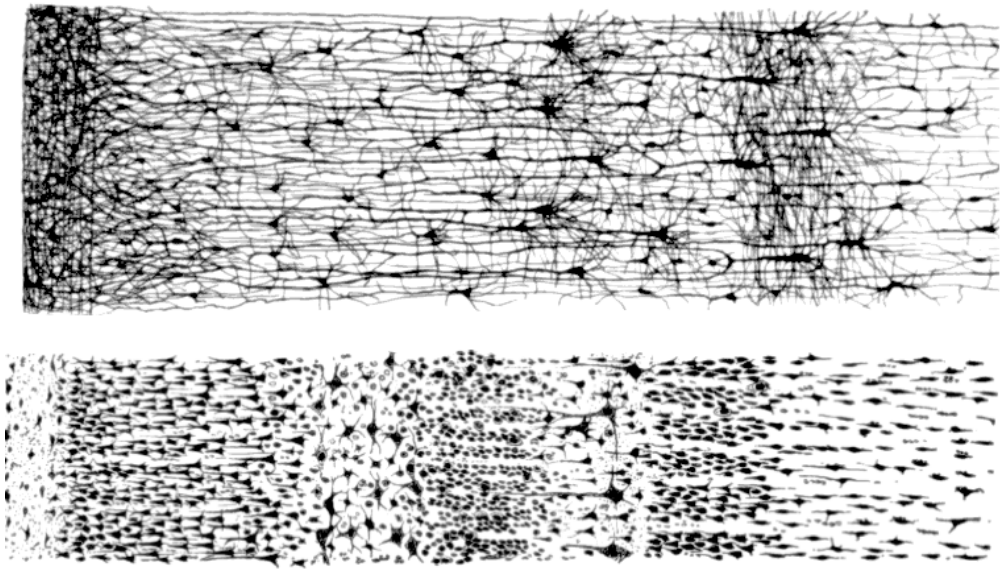


Figure 2.19: **The Human Cortex - A biological neural network.**. A vertical cross section of the human cortex between an adult (top) and 1.5 month old infant (bottom) showing a layer like structure with a change in depth (left to right). Source: Cajal [2020]

2.0.26.3 Applying The MLPRegressor To Observational Data

In the application of any type of machine aided algorithms, it is important to evaluate the results provided. In this section, the results of 12 years of data collected as part of the [CAPE VERDE CAMPAIGN] are shown (these contain measurements spanning the entirety of 12 years, which produce the clearest tests for the algorithm). A MLPRegressor of 10 hidden layers, and a hyperbolic tan (tanh) activation function is used **??**. Additionally, the limited-memory Broyden–Fletcher–Goldfarb–Shanno (l-BFGS) solver (a quasi-newton method which minimises the inverse of the Hessian matrix[15] to steer

[15]The hessian is square matrix of second-order partial derivatives of a scalar-valued function/field describing the local curvature of a function (of many variables).

through space and obtain a solution) and an adaptive learning rate[16] is used.

The input of the regressor is in the form of a month and an hour, to represent each measurement. This allows it to find not only daily trends but also seasonal trends within the data. Once trained the regressor is then used to predict a diurnal profile for each month based on the observational data provided. For simplicity $\log_{10}$ values of the concentrations obtained have been used. To validate the results, the predicted MLPRegressor line is compared to a transparent scatterplot for all the results. In addition to this, a boxplot showing the IQR, median and mean (green line) plotted alongside to evaluate the predictor output.

In providing the MLPRegressor with both month and hour inputs, the data is not only fitted hourly (a diurnal average) but also across the seasonal/monthly cycles. This accounts for the variation between years and datasets. Since $\log_{10}$ values of the concentrations are used, species such as ozone (Figure 2.20) which for the Cape Verde dataset (clean air) do not change more than one order of magnitude, the effects of neighbouring months, which shift the diurnal away from the mean (the green line on the boxplot), can be seen. However since this is overall a small change, and the diurnals lie within the interquartile range, they still provide an adequate approximation. NO (Figure 2.21) on the other hand has a concentration change of several orders of magnitude. Here a distinct daytime peak is seen and is centred around a seasonally consistent mean value of the data. Here the multi-magnitude change in concentration also provides an effective silhouette of the data to which we may compare the fitted line. Finally the plots of NO_2 and iso-Pentane (Figure 2.22-2.23) vary both in diurnal magnitude and seasonally. Within these plots, changes in the data in the January and December months produce deceptively misleading results. Here although the diurnals are not symmetric, they fit well within the median, mean and interquartile range values, as well as the general data silhouette behind them. This suggests that it is a property of the data that we are fitting, and not that the regressor is producing incorrect results. It is however noted that for a more accurate seasonal prediction, periodic boundary conditions should be employed in the training dataset, where an additional two months are added before January and after December. As only a single value estimate from the summer region will be taken, this does not affect the result accuracy.

[16]Each time the model improvement fails to decrease the learning loss, the learning rate is reduced by 1/5. This means smaller jumps are made towards the curve peak.

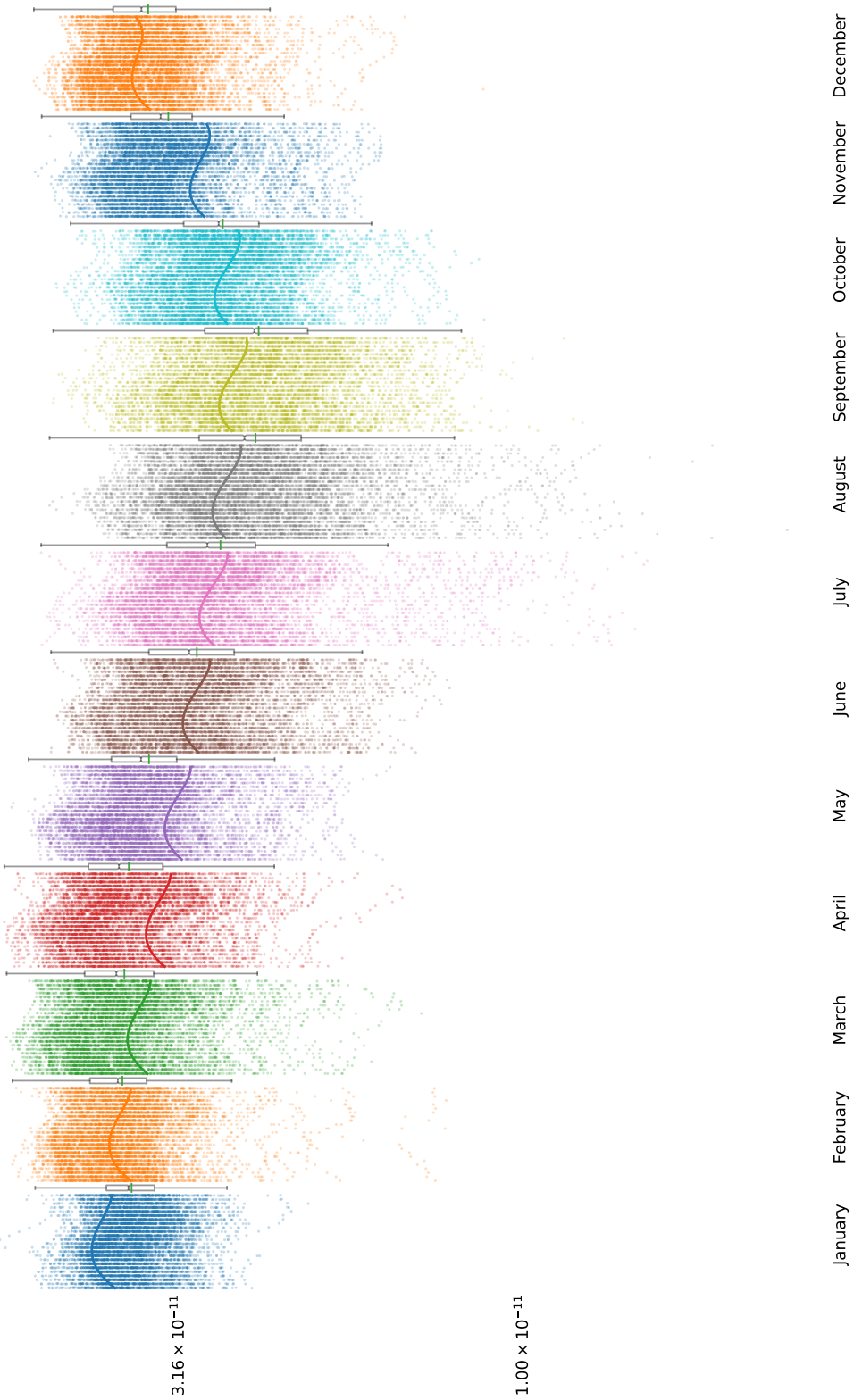


Figure 2.20: **Cape Verde MLP predicted and observational data of Ozone.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

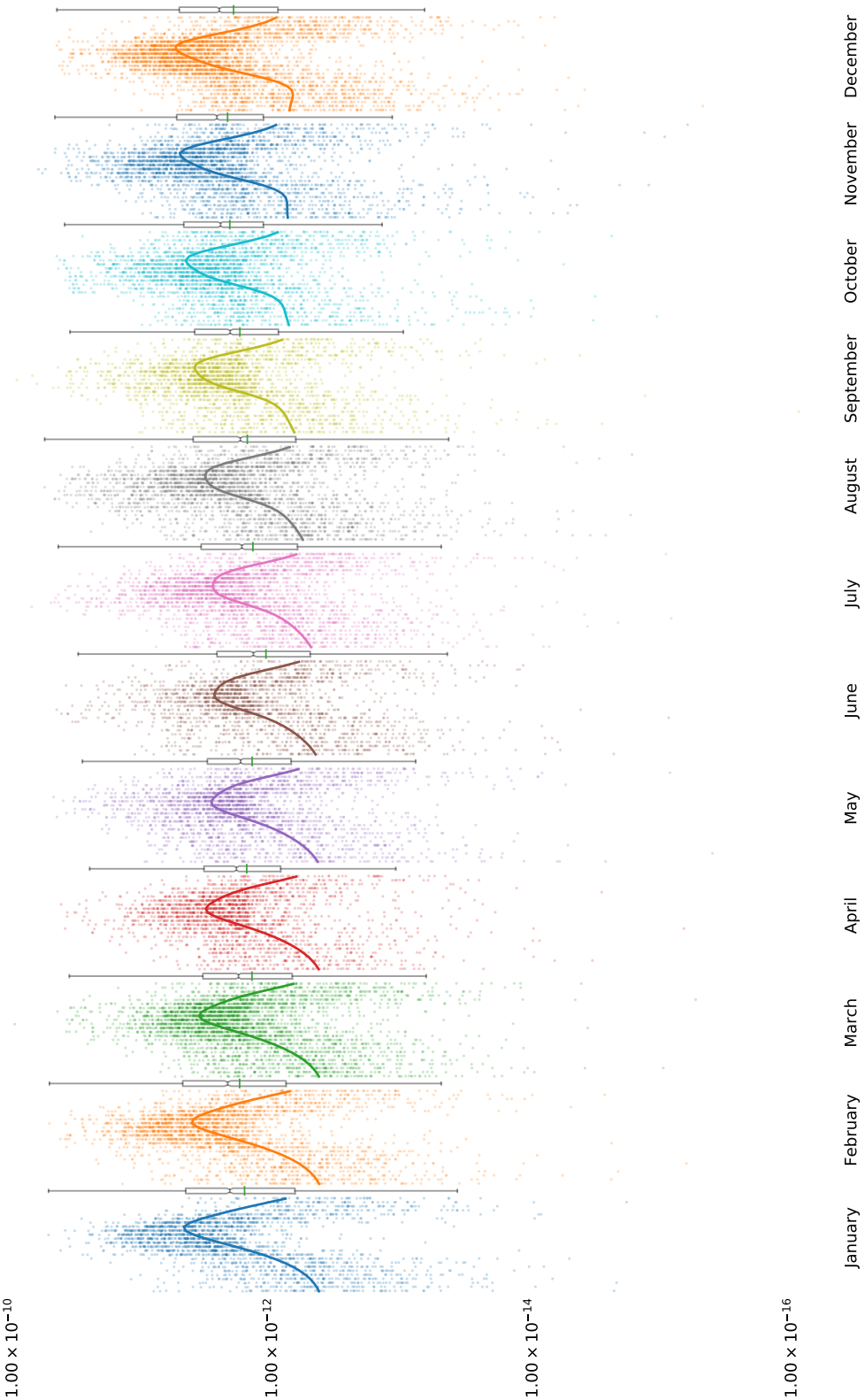


Figure 2.21: **Cape Verde MLP predicted and observational data of NO.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

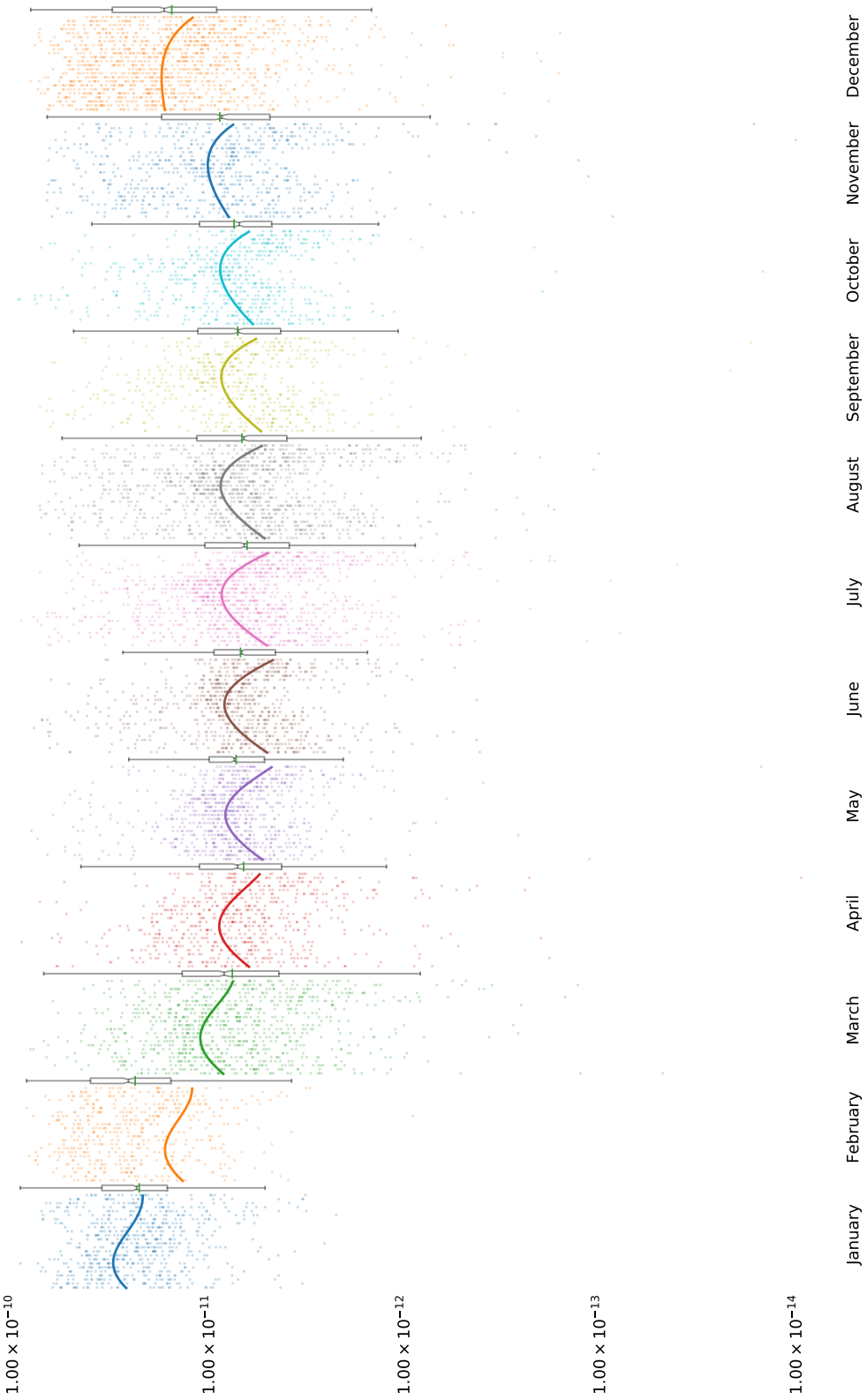


Figure 2.22: **Cape Verde MLP predicted and observational data of NO_2.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

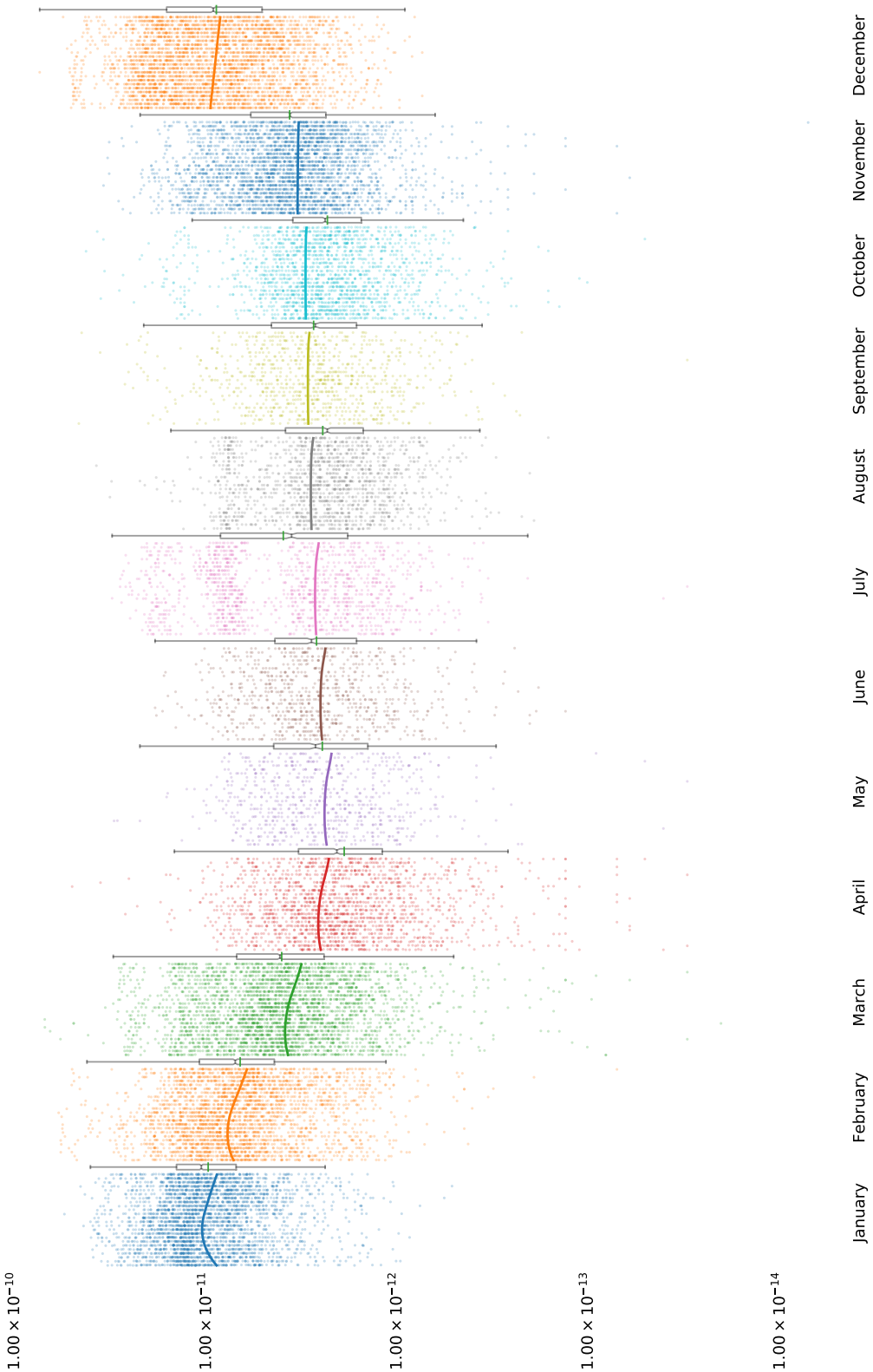


Figure 2.23: **Cape Verde MLP predicted and observational data of iso-Pentane.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

2.0.26.4 Model Initialisation Procedure

The aim is to generate a set of initiation concentrations which are representative of the different types of chemistry between environments. In this section, we are not interested in the exact concentration modelling for specific times or scenarios. Instead, we seek to generate representative of the processed chemistry under a range of conditions.

To do this species concentrations are extracted from an MLP regressor trained on observational data for each scenario. Each concentration is that of noon local time from the generated diurnal from summer observations at each location. This produces a monthly error of $\pm 2 months$ from June. As both nitrogen oxide and dioxide are supplied the total NO_x for each simulation are *not* constrained. The initial conditions are shown in Table 2.4.

In general observational measurements are not able to detect all the species presented within the MCM. This means that to be able to compare model scenarios, the chemistry must first be spun up. In propagating the chemistry forwards in time, primarily emitted and measured species are broken up forming the intermediate species which exist within a mechanism. To reach a steady-state, the model is initiated at noon and the observational concentrations are rest every 24 hours. For each diurnal, the fractional difference between the concentrations at each day are compared. If the difference between these is less than 0.001, the model is left to run unconstrained for 5 days (right of the dashed line in Figure 2.24-2.27). Model results are then taken after 3 days of unconstrained runs. The reason for this is that the total RO_2 concentration takes longer to stabilise in the polluted environments (London and Beijing). This falls into a periodic cycle beginning noon on the third day and can provide a representation of the processed chemistry within each environment.

NOTE: It should be noted that some of the concentration plots may appear to lose their diurnal dependability. This may be attributed to the changing order of magnitude of the concentrations, and that the species are still responding as expected.

2.0.26.5 Extracting The Required Results

Model diagnostics such as concentration and the net flux passing through a species may be extracted directly from the DSMACC box model. These provide the baseline comparison and can be directly compared to the graph metrics. Species concentration tells us the abundance of different species, and the net-flux tells us how fast this is changing in time.

As some species may have a fast inwards and outwards flux (low net-flux), the absolute flux is also included. Finally, the sensitivity of each species for other species is also extracted (the jacobian matrix). This serves to not only generate the graph used to represent the chemistry, (Subsection 2.0.23)

but also to identify the overall influence a species has on others in the network. This can be calculated by taking the net sum of the influence a species has on every other from the Jacobian. This is analogous to calculating the outdegree of a node in the jacobian network.

Species	Beijing(APHH)	Borneo(OP3)	London(ClearFlo)	CapeVerde
LAT	39.9	0.96	51.0	16.5
LON	116.3	114.5	0.00	23.4
O_3	6.883e-08	8.939e-09	3.819e-08	2.629e-11
NO	1.660e-09	2.668e-14	2.350e-09	2.358e-12
NO_2	1.226e-08	1.081e-13	7.445e-09	8.447e-12
HCHO	4.472e-09		1.119e-08	
C_2H_6	3.163e-09	7.315e-10	2.133e-09	4.539e-10
C_2H_4	1.004e-09	1.152e-10	4.893e-10	2.481e-11
C_3H_8	3.019e-09	1.924e-10	1.128e-09	1.728e-11
C_3H_6	1.335e-10	1.333e-11	1.784e-10	9.343e-12
IC_4H_{10}	6.412e-10	8.742e-11	5.142e-10	2.486e-12
NC_4H_{10}	1.593e-09	5.698e-11	1.058e-09	4.481e-12
C_2H_2	1.058e-09	1.825e-10	3.018e-10	1.848e-11
TBUT2ENE	4.198e-11		1.815e-11	
CBUT2ENE	4.454e-11		1.305e-11	
IC_5H_{12}	1.047e-09	2.883e-11	7.424e-10	3.470e-12
NC_5H_{12}	4.650e-10	2.090e-11	2.792e-10	2.513e-12
TPENT2ENE	3.939e-11			
CPENT2ENE	3.982e-11			
NC_6H_{14}	2.057e-10	6.437e-12	6.357e-11	
C_5H_8	7.134e-10	1.957e-09	1.640e-10	
NC_7H_{16}	7.905e-11		5.222e-11	
BENZENE	4.045e-10		1.137e-10	7.682e-12
NC_8H_{18}	3.091e-11		1.442e-11	
TOLUENE	6.767e-10		3.205e-10	3.121e-12
EBENZ	3.115e-10		6.017e-11	
OXYL	1.677e-10		5.049e-11	
CH_3CHO	4.783e-10		4.095e-09	
C_2H_5OH	4.655e-09		3.125e-09	
CH_3COCH_3	3.328e-09		2.924e-09	
NC_9H_{20}	1.336e-11		7.922e-11	
$NC_{10}H_{22}$	1.062e-12		1.602e-10	
α-PINENE[17]	7.341e-11	15e-11	1.105e-10	
LIMONENE	5.836e-11	1.351e-10	3.566e-11	
$PXYL^+MXYL$[18]	4.943e-10			
IPBENZ	4.567e-10			
PBENZ	3.996e-10			
HONO	6.479e-10		4.109e-10	
MACR		6.948e-11	1.862e-11	
$PENT_1ENE$			2.383e-11	
MVK			2.091e-11	
NPROPOL			2.883e-10	
NBUTOL			4.535e-10	
STYRENE			2.241e-11	
MEK			5.494e-11	
C_3H_7CHO			9.534e-12	
C_4H_9CHO			1.865e-11	
$C_5H_{11}CHO$			1.201e-11	
CYHEXONE			9.790e-12	
BENZAL			1.510e-11	
PAN			1.791e-10	

Table 2.4: The initial conditions created from the MLPRegressor prediction of observational data. Although not specified the concentration for methane is set by the model at 1770ppb.

[18]This is written as ?-pinene in the merged CEDA dataset for the Borneo OP3 campaign. This is due to character conversion errors.

[18]The concentration for these is split evenly between both species

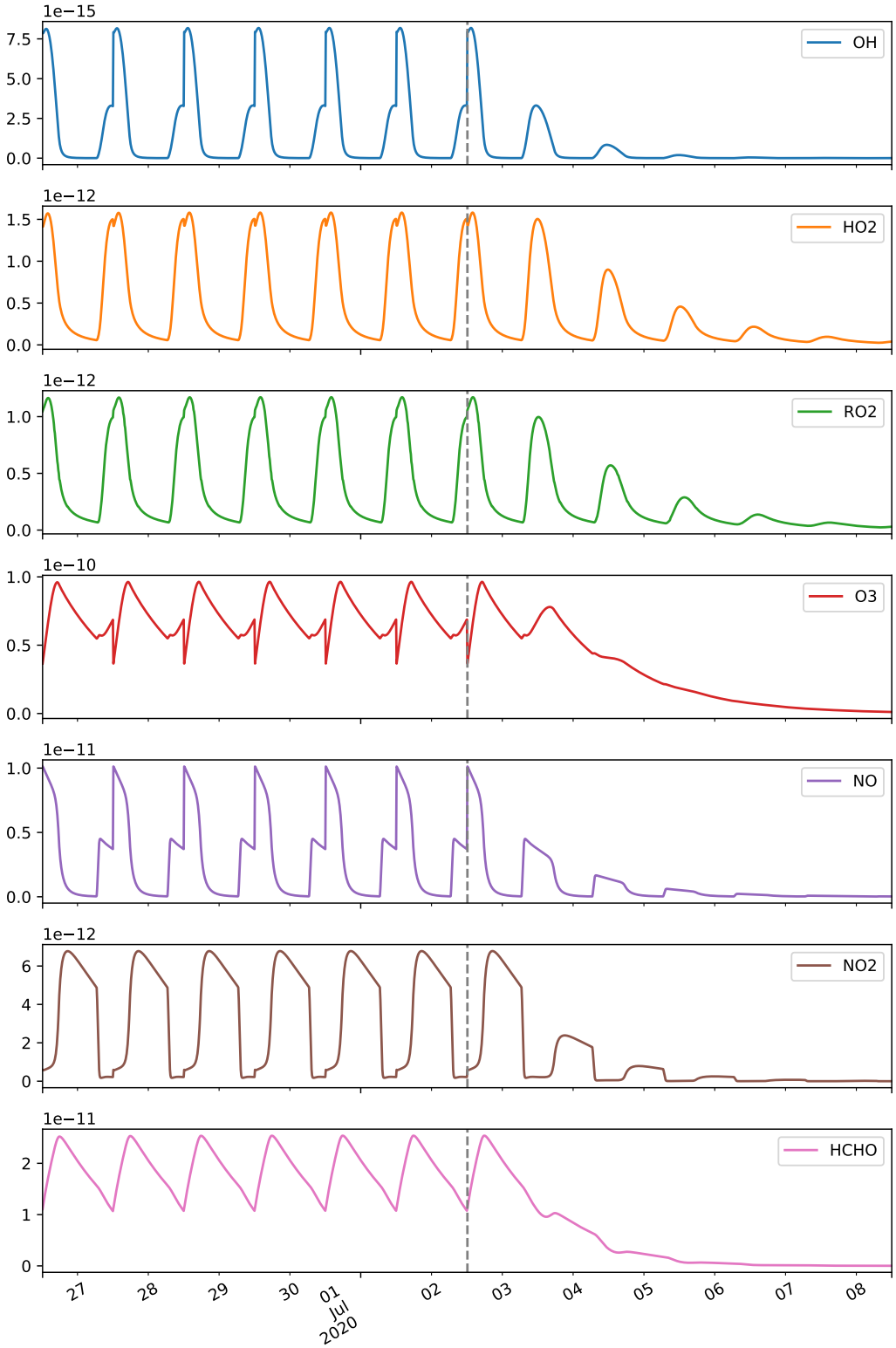


Figure 2.24: **The concentration profile for CapeVerde.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

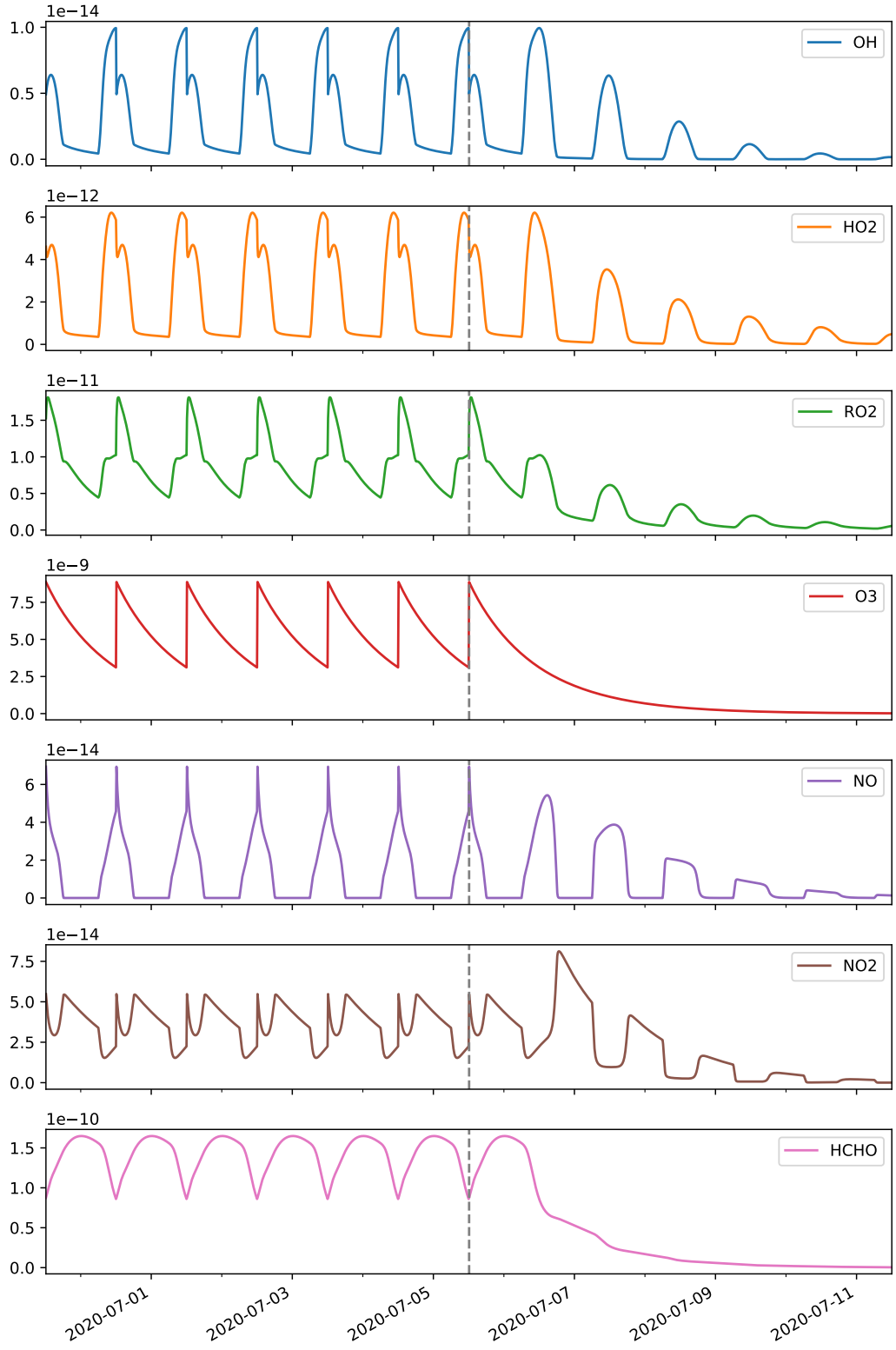


Figure 2.25: **The concentration profile for Borneo.**This shows a the change in concentration over time for HO_x, NO_x, Ozone and RO_2 species for a simulation run generated by the MLPRegressor. Left of the dashed line shows the last 6 days of spinup, where the initial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

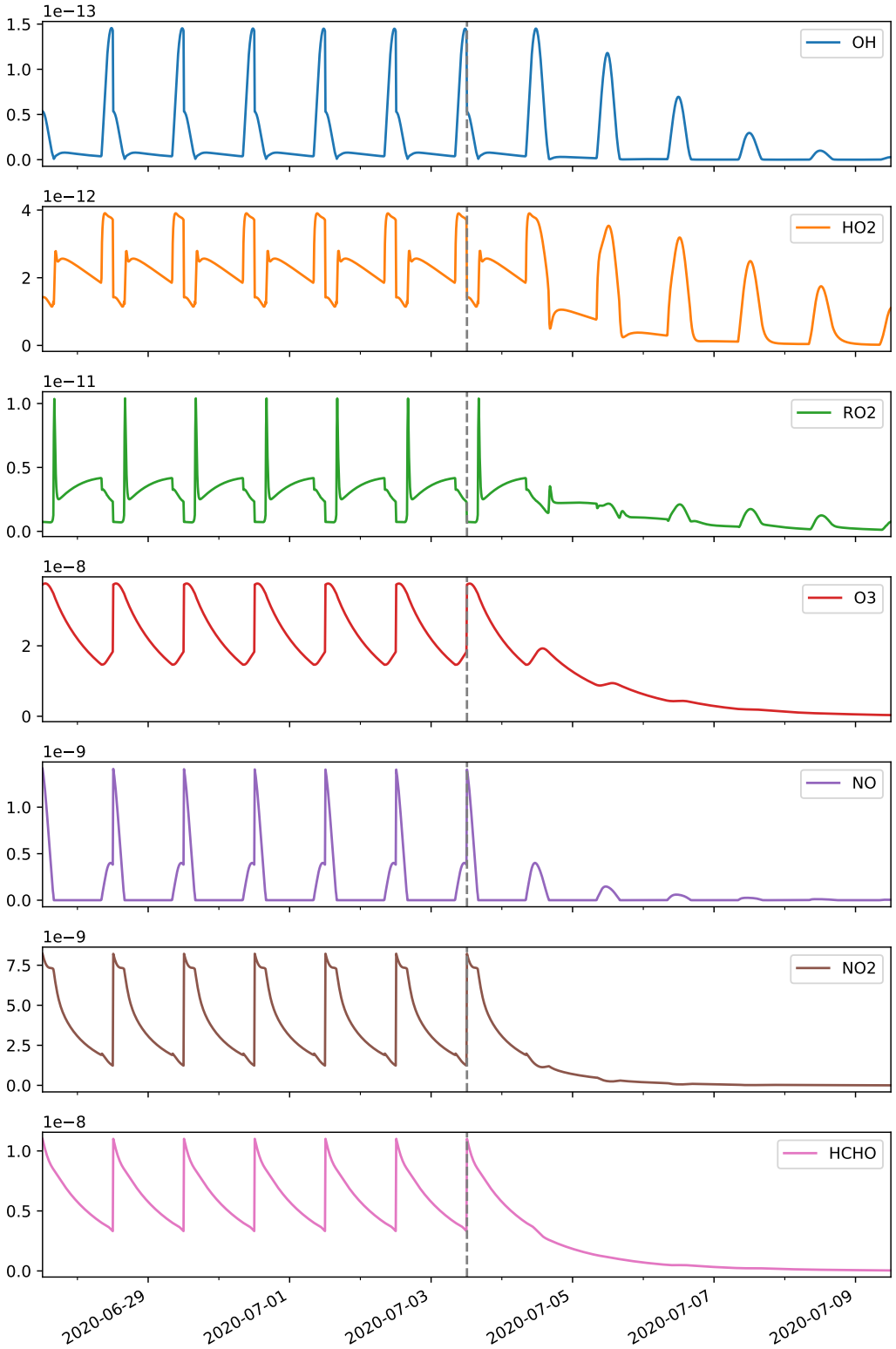


Figure 2.26: **The concentration profile for London.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

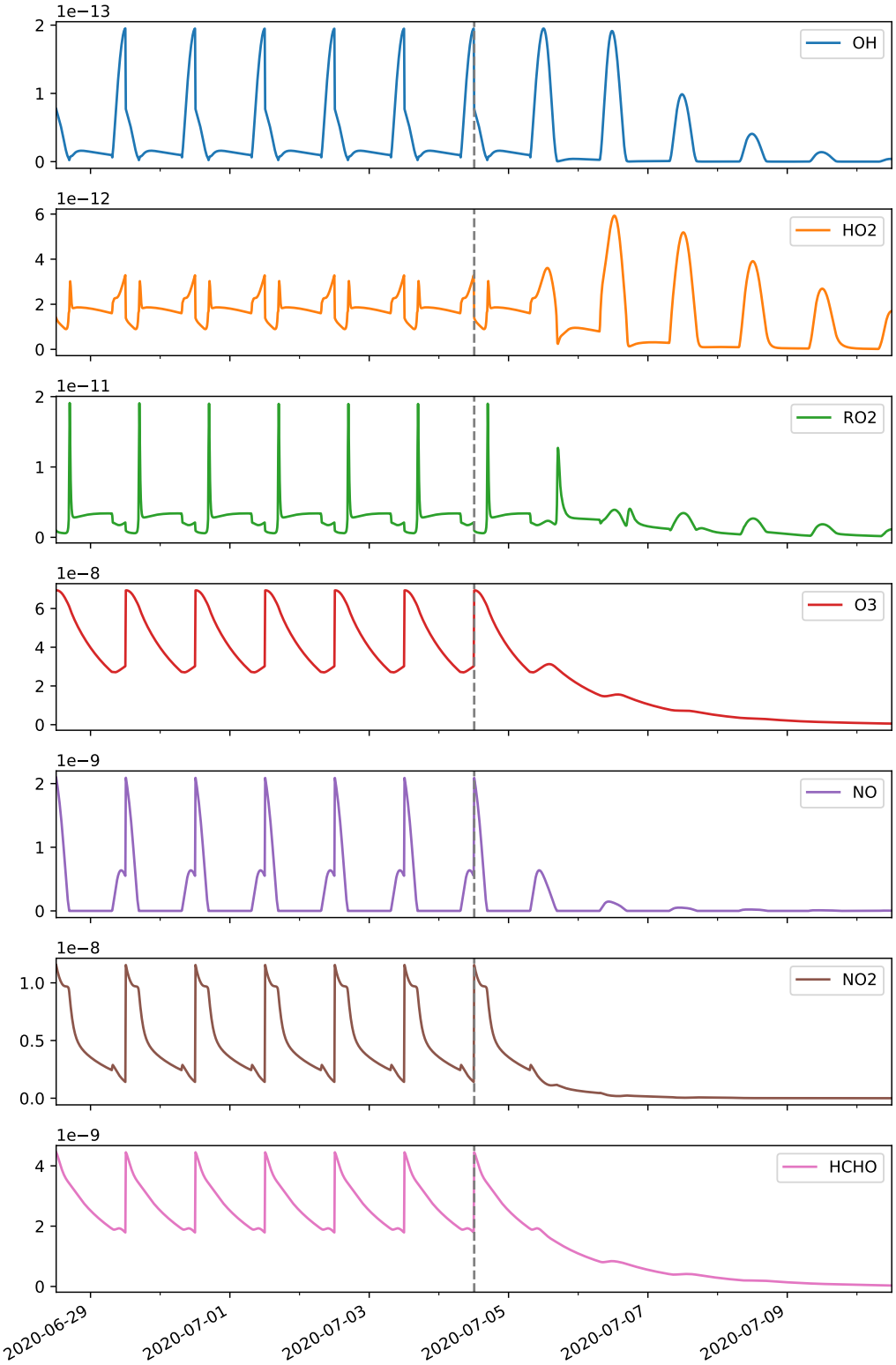


Figure 2.27: **The concentration profile for Beijing.**This shows a the change in concentration over time for HO_x, NO_x, Ozone and RO_2 species for a simulation run generated by the MLPRegressor. Left of the dashed line shows the last 6 days of spinup, where the initial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

2.0.26.6 Unifying The Results

Each metric provides a different range in which it ranks the importance of a node. To account for this all results are scaled to the range {0,1}, where 1 is the highest. Entries, where the results span several orders of magnitude (e.g. concentration, flux, influence), are flattened using the $\log_{10}$ scale before being normalised.

2.0.27 Comparing Results

This subsection juxtaposes the use of traditional model diagnostic methods against a selection of graph metrics. As there are several thousand species within each simulation run, the keyword extraction algorithm Term Frequency - Inverse Document Frequency (TF-IDF), is used to identify the top most prominent species for each metric (traditional and graph). From this, the 10 highest-ranking species from each category are collated into a single diagram for comparison.

2.0.27.1 What Is TF-IDF

TF-IDF is a numerical statistic used in text natural language processing and text mining. It is designed to identify the importance of a word concerning its context.

It provides a value for the frequency a word appears within a document, offset by the number of times it appears in other documents within the corpus - It is for this reason that 83% of text recommender systems in digital libraries use TF-IDF, [Beel et al., 2016].

In [Ellis, 2019] I applied this to the chapters of Frankenstein and found the keywords extracted almost exactly replicated those from the synoptic description of the novel. Although TF-IDF is a text mining procedure, the algorithm itself is mathematical, meaning that it may be applied to our diagnostic dataset. The working of the algorithm is discussed below.

Term Frequency

The TF from the algorithm name stands for term frequency. This is an analysis of the number of times a word exists within a dataset. There are several ways in which this can be done, these are:

- **Raw Count** - The *number of times* a word exists within the document.
- **Boolean/Logistic** - $True$ if the word exists, false otherwise.
- **Adjusted for Document Length** - $word\ frequency/total\ number\ of\ words$

- **Log Scaled** - $\log(1 + frequency)$

As the scaled values for each item are taken, we can liken our results to the 'Adjusted for Document length' equation and use the scaled ranking value for each group respectively.

Inverse Document Frequency

Inverse document frequency tells us how much information a word provides concerning a certain context. Whilst a word may be used extensively throughout the corpus (i.e. term frequency) it is often that we are interested in words which are only frequent within a specific document. This is one of the reasons TF-IDF is useful in the extraction of keywords from a document.

The inverse frequency of a word is usually calculated as the log of the fraction of documents N against the number of documents the word appears in D_f, Equation 2.14.

$$IDF = \log(\frac{N}{D_f}) \tag{2.14}$$

If required, changes can be made to produce results which show a better representation of words which are important for all documents (probabilistic, Equation 2.15) or individually (smooth, Equation 2.16). However in looking at Figure 2.28, it can be seen that the basic IDF formula mentioned has a limit of zero the greater the document frequency (D_f), which makes it easy to normalise against - i.e. divide by 2 as this is the value tended to if the document frequency tends to 0.

$$IDF_{prob} = \log(\frac{N - D_f}{D_f}) \tag{2.15}$$

$$IDF_{smooth} = \log(\frac{N}{1 + D_f}) + 1 \tag{2.16}$$

To complete the TF-IDF equation, the term frequency and inverse document frequency terms are multiplied together.

Applying TF-IDF to chemical metrics

To identify metrics selection criteria, we seek only species which are important only in that category. To do this the TF-IDF algorithm can be adapted for use with the graph metric output. Here 'Term Frequency' corresponds to the number of times a value appears within the body of a document and

can be seen as the scaled {0,1} metric output. This is then divided by the log of the 'Inverse Document Frequency' with D_f being the sum of values across all the metrics. This makes the TF-IDF equation:

$$TF.IDF = metric\_value\ .\ \log(\frac{N_o\ documents}{\Sigma_\forall\ metric\_values}) \tag{2.17}$$

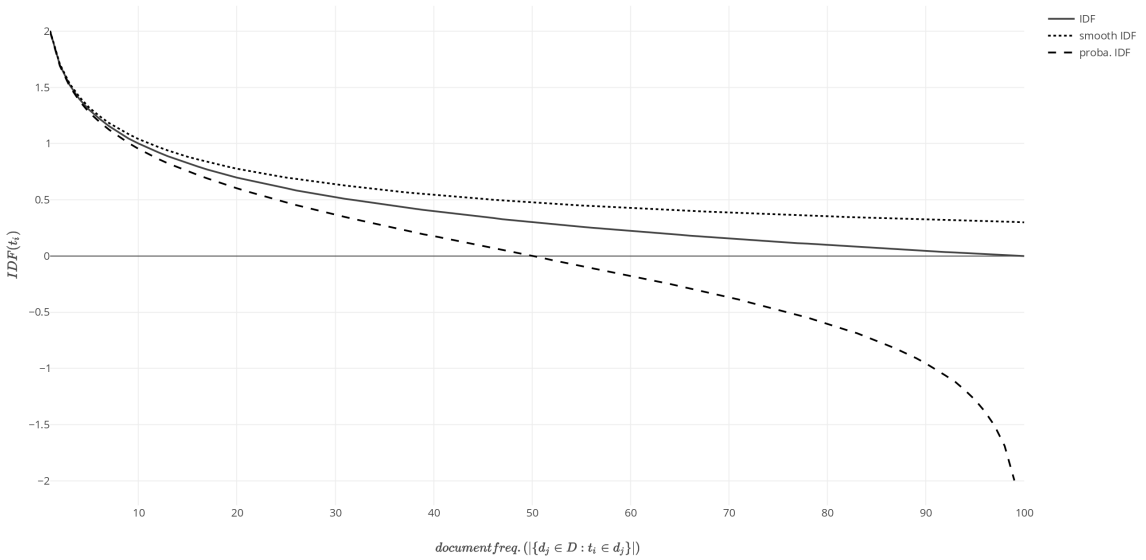


Figure 2.28: **The different IDF outputs.** A plot showing Inverse Document Frequency profiles against Document Frequency. This shows that the probabilistic IDF highlights words that are more important across all items, whilst the smooth IDF shows files which are more important individually. The general IDF (which is used) produces a result starting at 2 and tending to zero. This provides the best response and can easily be scaled between the range of [0,1] by dividing the output by 2. Source: [Mquantin, 2020]

2.0.27.2 Metric Comparison

This section aims to compare the efficiency of graph metrics against a list of traditional methods. To do this the use of a bivariate colourmap (Figure 2.29) is used. Each figure consists of a red-hued image/heatmap representing the scaled values {0,1}:{white, red} for each of the individual columns. As each simulation contains thousands of species, only the top 10 species from each column/category are selected. These are then sorted by the average sum of their closeness, betweenness and page-rank values (blue column). Superimposed on this reds-only heatmap is a blue heatmap representing the average sum of the three metrics for comparison. Such a method allows for the comparison of individual values against an approximation of species importance, by the sum of graph metrics - allowing an easy categorisation of the data:

- **Purple** - This value is high in both the individual category and the metric sum.

- **Red** - This value is high for the individual category but not the metric sum.

- **Blue** - This value is high for the metric sum but not the individual category.

- **White** - This value is low for all categories.

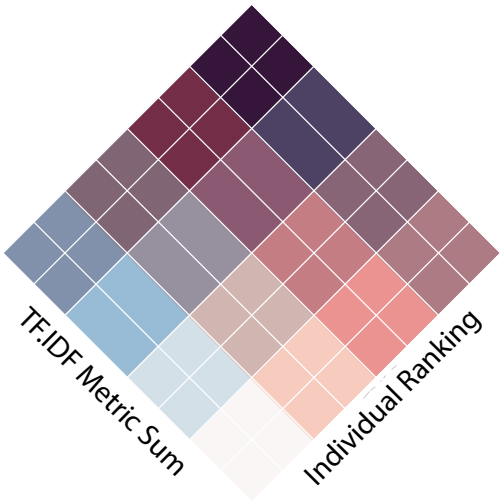


Figure 2.29: **The bivariate colourplot key.**

2.0.27.3 Individual Categories

Individual categories are split between traditional metrics and graph centrality metrics. To represent the importance of a species the following values may be extracted through the use of a simple box model:

- **Concentration** - This describes the abundance of a species within the atmosphere.

- **Net Flux** - This describes the rate of net (absolute) change of concentration over time for a species.

- **Absolute Flux** - Some species may have a large flux going through them (production and loss), resulting in a small net flux. This sums the production and loss fluxes.

- **Influence** - Influence is the total magnitude of an effect that changing a species concentration by 1% would have on other species within the network. Since the graph is generated using the Jacobian matrix, an alternative method for calculating this can be by calculating the total out-degree of a node.

The importance of a species is then compared through the use of three of the most common centrality metrics. These are:

- **Centrality** - This describes how easily information from one node can be disseminated to all other nodes.
- **Betweenness** - This describes the number of shortest paths (fastest fluxes/greatest influences) that are routed between nodes adjacent to our chosen node. Species with a high betweenness hold a brokering position and can act as a bottleneck between different groups of chemistry.
- **PageRank** - PageRank looks at the flow in a system. It ranks nodes not only on the number of species it reacts with but also the importance of the species it has reacted with.

Finally, the 'Metric Sum' is the sum of all the metric values scaled between 1 and zero (the mean).

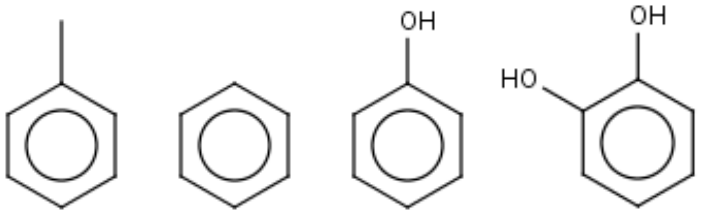
2.0.28 Senario Analysis

In selecting the top 10 ranking species for each category it is possible to examine if the importance of a species with centrality metrics varies from the results suggested by traditional metrics. In this subsection, we explore the TF-IDF rankings of each metric and use this to decide if species importance is local to a specific metric. We look at what species are highlighted by each scenario and compare them against the primary emitted species shown in Table 2.4. Finally, we compare the total metric sum against the traditional metrics of concentration and flux and compare the correlation.

Cape Verde

The initial conditions for Cape Verde have low levels of NO_x and ozone. The chemistry is split between aromatics and small alkanes. The aromatic species are of a similar magnitude to the alkanes. Many of the aromatic products are shown to be important in Figure 2.31, which may be due to the larger aromatic species <break down?> potential (they have more carbons to form bonds with). Using this it can be seen that many of the species highlighted are products of Toluene, Benzene, Phenol and Catechol (the latter of which are produced by adding an alcohol group to a Benzene ring - Figure 2.30). These are most likely emitted either from mainland Africa or through ship emissions and are important indicators of how processed the chemistry of the atmosphere is. Benzene and Toluene are usually emitted at a ratio of 1:4 respectively and the same rate. Since Toluene reacts at a much faster rate, a change to this ratio allows tells us how much the chemistry within a system has changed. It may be suggested both from the initial conditions and the metrics that the chemistry is one which has been transported to the island, rather than emitted there. In addition to the aromatics, many of

the primary emitted alkanes, and their products, have been highlighted. These tend to be unreactive (and thus long-lived) which can be seen by their low betweenness[19] values - they are unlikely to act as a fast-reacting proxy between two other species. The change in colour between the metrics suggests that for the Cape Verde Mechanism important species are not often central (easy to get to - closeness centrality) from all other species. The selected species ranking the highest closeness (Propane-Pentane) do not seem to be ranked important as part of betweenness or page rank which results in the red colour for the plot (low overall metric sum). Species ranked high by the PageRank algorithm do not have a high betweenness or net flux value, but do have large absolute flux. This suggests that although they may not have the fastest fluxes going through them (low betweenness), they act as an intermediate reaction for other chemistry where they are produced and lost at a similar rate (low net flux, high absolute flux.).



(a) Toluene (b) Benzene (c) Phenol (d) Catechol

Figure 2.30: **Chemical structures of the 4 most common type of aromatic species in cape verde.**

Borneo

The Borneo dataset, through the nature of being located in a rainforest, contains no benzene ring based aromatics. From its initial conditions the simulation begins with a higher level of Ozone, High Isoprene (C_5H_8), and moderate amounts of Acetylene (C_2H_2), α-pinene, and limonene (both ($C_{10}H_{16}$)). Figure 2.32 shows a very large of the rainforest chemistry is dominated by terpenes (mainly Isoprene) products. Unlike Cape Verde, these all have a high concentration, net-flux and absolute flux. This suggests that the products act as intermediate species for the chemistry and are both produced and lost at a high rate. Much of these species have a high closeness and a high page rank, suggesting that the centre of the Borneo network is very close-knit, and the well connected to species of importance. The only outlier to this is $CISOPAO_2$ which is has fast reactions flowing through it and has important connections (it is only a couple of steps away from Isoprene) but does not have a high closeness centrality. This suggests that it is located as part of a terpene branch but not at a highly pivotal position. The uniformly distributed colour gradient for betweenness suggests

[19]Most of the species with low betweenness values are a product of Ethane (C_2H_6), Propane (C_3H_8), Butane (C_4H_8) and Pentane (C_5H_{12})

that there are many possible reaction routes a species may undergo before being converted into carbon dioxide and water. The exception to this is $C_{517}CHO$ which has 14 precursors and only 2 products (a bottleneck / pivotal position), resulting in the highest betweenness value of the network.

London

The London dataset contains a mix of anthropogenic and biogenic aromatics and long-chain alkanes. Similar to Cape Verde we have a section of alkanes which have a low overall metric sum, with a small value for closeness and page rank. Combined with their high net flux, absolute flux and influence values, this suggests that they have a moderate directional flux, most likely influencing the production of many other species at a consistent rate. In addition to these, we have species with a moderate closeness but a high betweenness. These are often species such as formaldehyde (HCHO), glyoxal (C_2O_2) and acetaldehyde (CH_3CO_3) which can serve as tracers for fast photolytic reactions. This is because on the graph structure (**??**) they sit between the dense centre of the network (high closeness) and the branches formed from each primary emitted species (low closeness). Their high connection density and importance in the network is also picked up by the page rank algorithm. Other species with high betweenness and a low centrality are the monoterpenes limonene and α pinene, as well as hexane (NC_6H_{14}) and butane products. These are (or are close to) primary emitted species and therefore have a low closeness. Since this also means that much of the chemistry originates with them, the outward 'flow' of information also results in a lower page rank value.

Beijing

Similar to London, the fast photochemical tracers are identified, although some have a slightly lower flux between them (betweenness) and page rank values. This suggests that the network structure or weightings may have shifted slightly, creating more links, or importance in a specific branch of chemistry. Additionally, their overall metric sum is lower. Glyoxal, Methyl Vinyl Ketone (MVK) and their associated criegee configurations all feature heavily in the middle of Figure 2.34. These are important as they represent the fast chemistry formed by both the anthropogenic and biogenic chemistry that is within the simulation. These tend to have a high closeness and page rank centrality, a pattern that is also seen with the long-chain alkane products from Octane (NC_8H_{18}), Hexane (NC_6H_{14}) and isoprene.

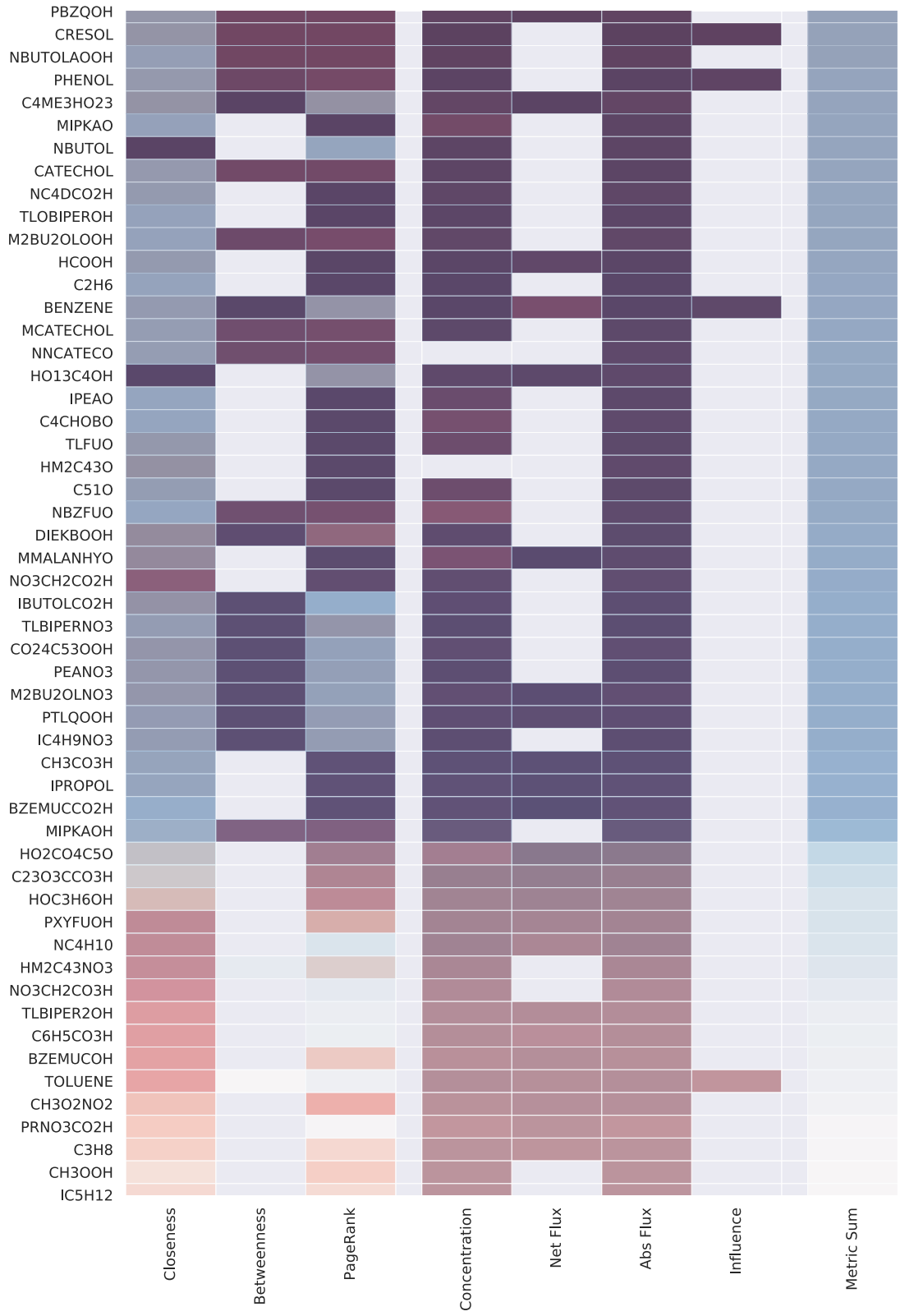


Figure 2.31: **A bivariate heatmap comparison of Cape Verde.**

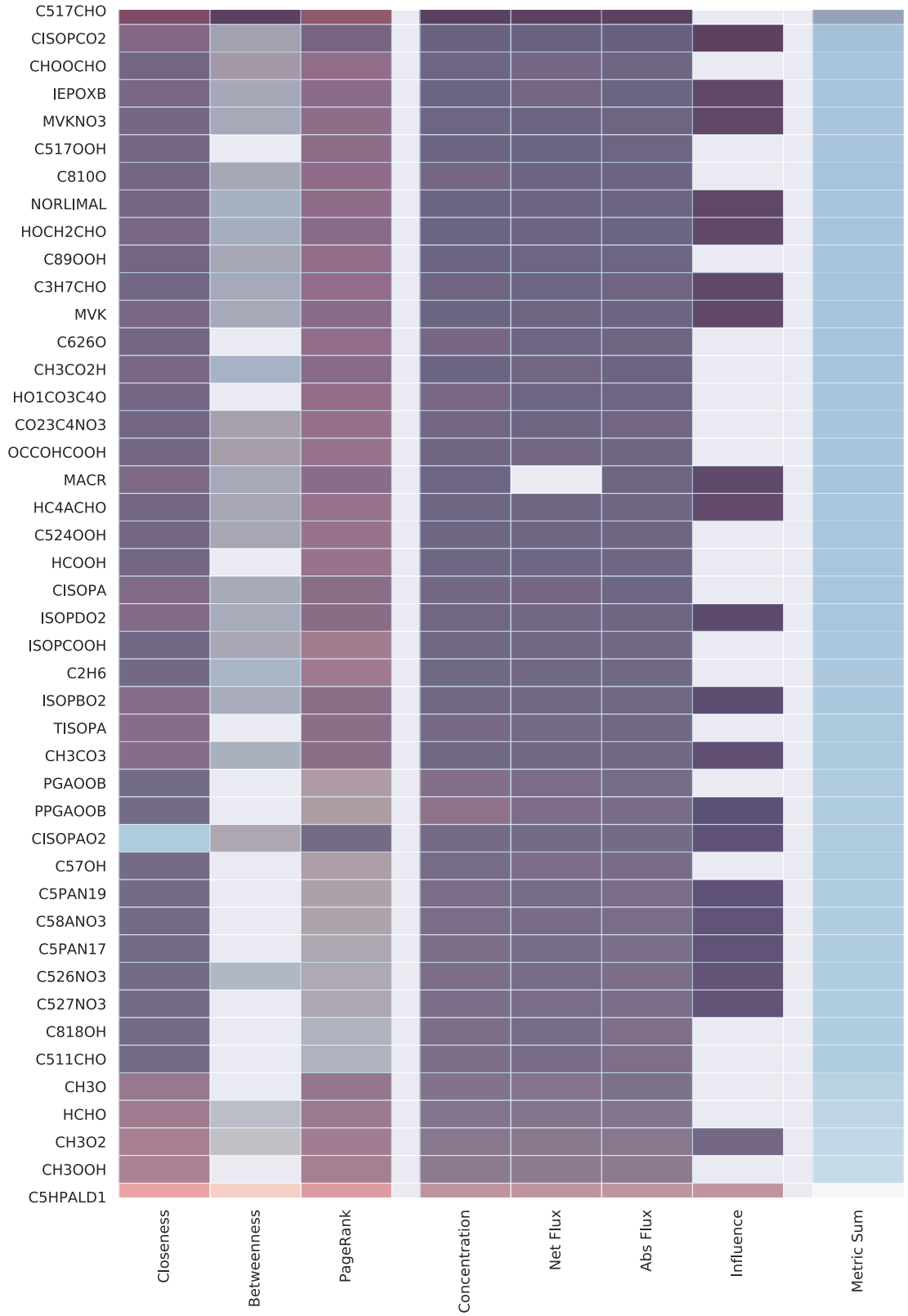


Figure 2.32: **A bivariate heatmap comparison of Borneo.**

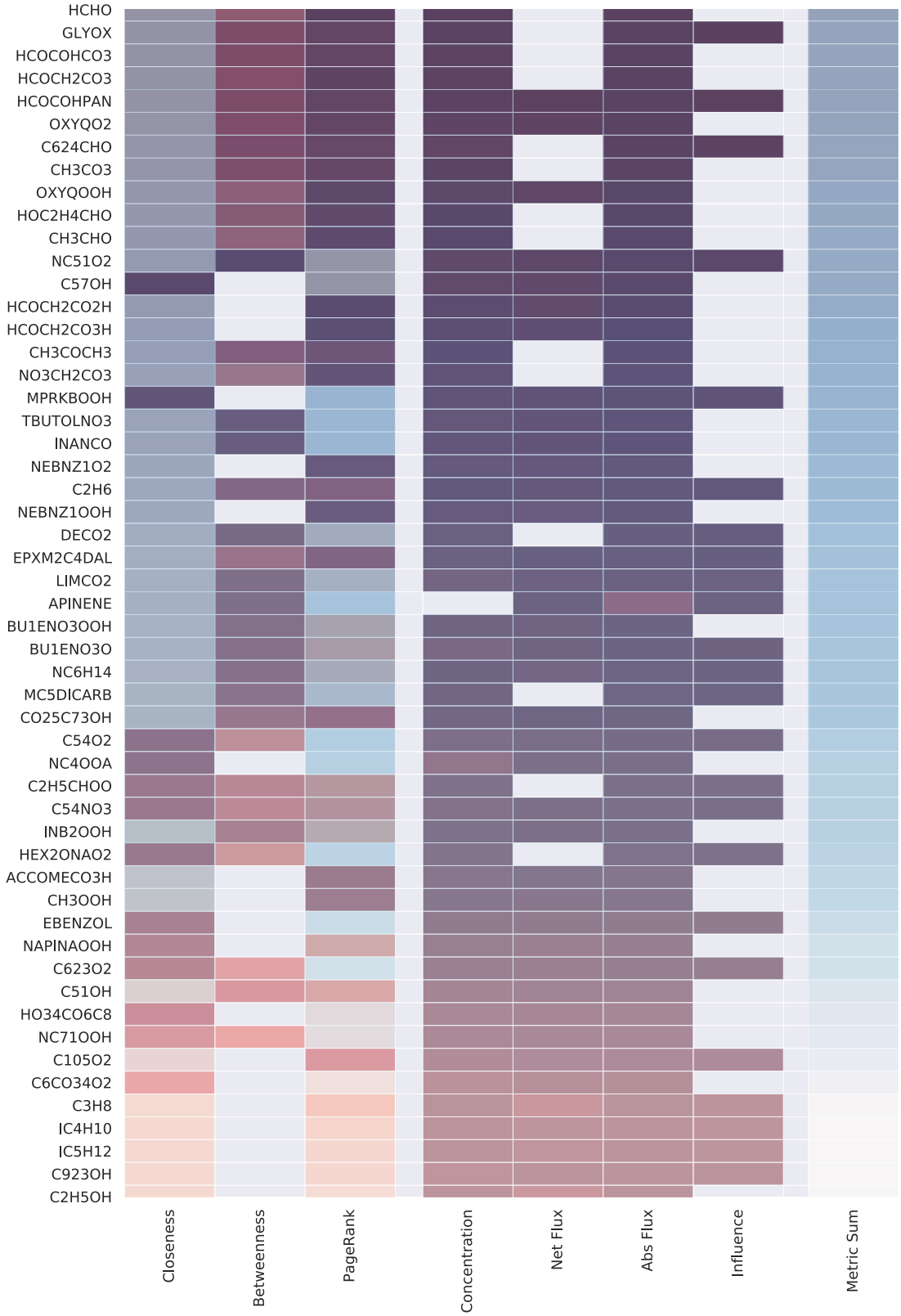


Figure 2.33: **A bivariate heatmap comparison of London.**

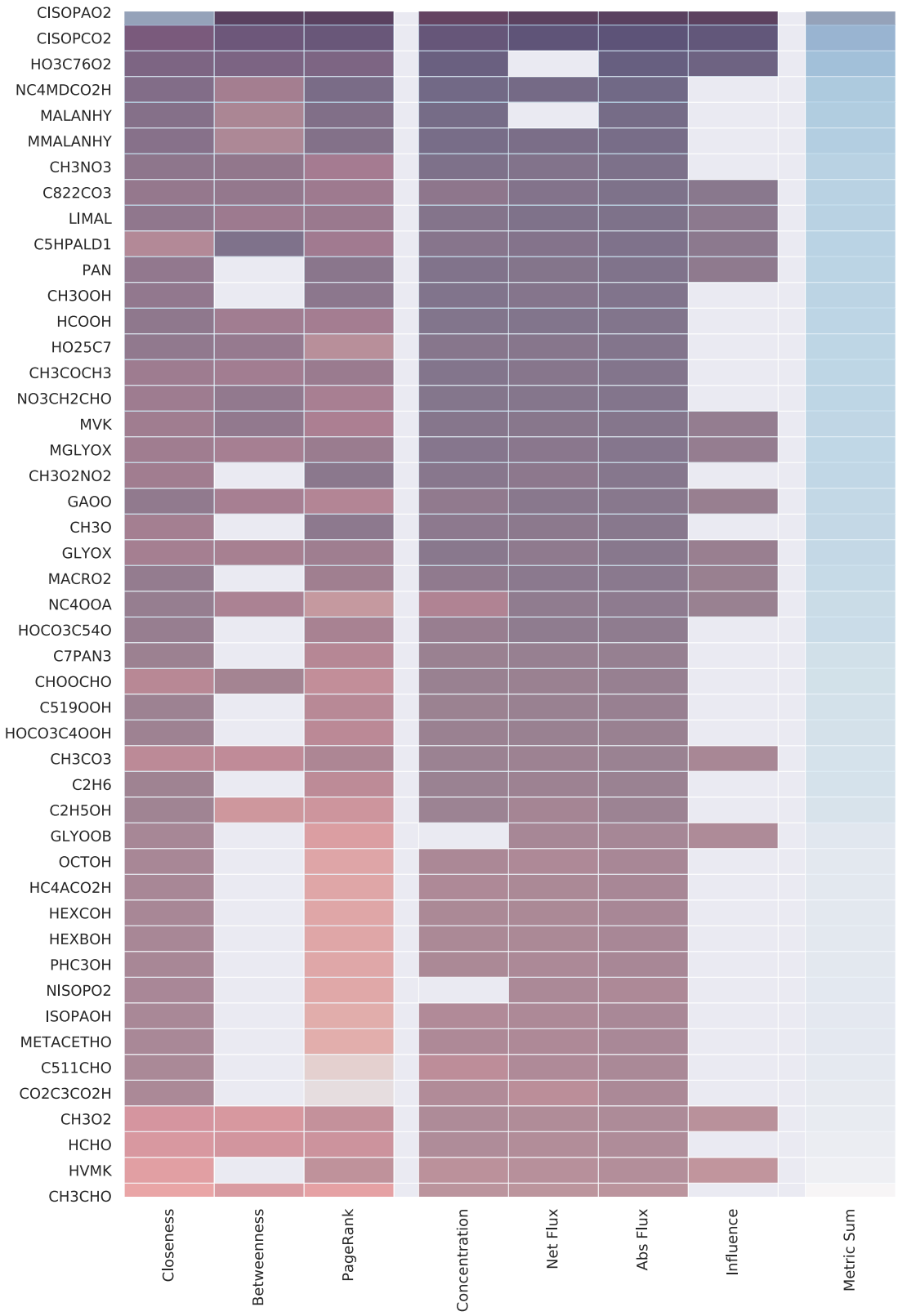


Figure 2.34: **A bivariate heatmap comparison of Beijing.**

Ability To Match Traditional Metrics

All graph construction Colours - all purple, suggests a general agreement photochemical tracers

2.0.29 Providing An Overall Overview Using The TF-IDF And The Metric Sum.

In the previous section, it was shown that centrality metrics can be used to complement the use of traditional metrics in the analysis of the chemical network. As each metric represents a different aspect of importance, should a single ranking value for a node be required, it is possible to take the average sum of all three metric values. Looking at Figure 2.31-2.34 it is possible to see similar trends in colour gradient between the purples of the traditional metrics of flux and concentration with the total metric sum (the blue column). This suggests that it is possible to compare each scenario with the use of the metric sum.

In selecting the ten highest-ranking species from the mean centrality metric table for each simulation, Table 2.5 can be created. Unlike the previous method, we are now looking at species which are important across all metrics in a simulation. For Borneo species produced from Heptane, Hexane, Isoprene and Limonene are seen as important. Cape Verde, similar to before, has a selection of Benzine related products such as Phenolic and Catecholic compounds. Beijing consists mainly of Quinones and Dialdehydes which are both derivatives of Benzene. London again has Benzine related compounds, mixed with the fast photochemical indicators, which were also ranked highly in Figure 2.33. Looking at the highest-ranking sum (Nan-mean), it is seen that isoprene, hept/hexane and glyoxal products highlighted as the most consistently important across all four simulations.

	London	Cape Verde	Beijing	Borneo	Nan-Mean
0	HCHO	PBZQOH	PTLQONE	C622OH	CISOPCO2
1	CH3CHO	PHENOL	PBZQONE	C923OH	CISOPAO2
2	C5CO14OOH	C24O3CCO2H	HOHOC4DIAL	C54OH	C517CHO
3	PBZQOOH	NBZFUONE	MNNCATCOOH	HO2C4OH	HO2C6O2
4	MALANHY	TLBIPERNO3	C6H5CO3H	C624OH	HCOCH2CO3
5	CH3CO3	BZBIPERO	EPXDLPAN	HEXAOH	C717O2
6	C57OH	TLEMUCCO2H	C5DIALO	C822CO2H	HCOCOHCO3
7	C624CHO	BZEMUCCO2H	NBZFUOOH	MACROHOOH	HOCH2CH2O2
8	GLYOX	PTLQO	TLBIPEROOH	HO14CO2C4	HOC2H4CHO
9	HCOCOHCO3	NNCATECO	NCRESOOH	C624CO2H	C626CHO

Table 2.5: **A table of the top 10 ranked species for each simulation.** Only species that exist within atleast 3 out of the 4 simulation are used. The Nan-Mean takes the mean of all available data, ignoring runs where a species is not present.

A note on finding the precursors

Graphs are also useful in the back navigation of a network. It is possible to discover the most probable

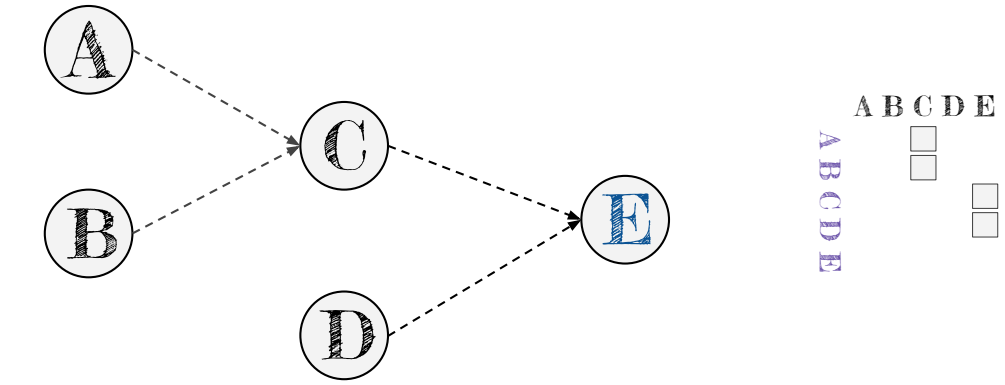
primary emitted species (nodes with no in-degree) by comparing the shortest path lengths for all primary emitted species (not including inorganic species). Here the primary emitted species with the smallest number of connections are often the most likely source.

2.0.30 Calculating production sensitivity using personalised page rank.

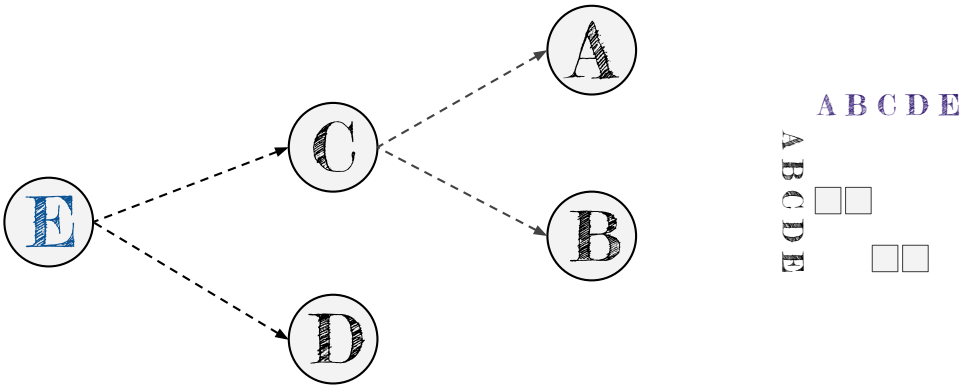
In the calculation of the PageRank result by solving for the eigenvalues and vectors of the google matrix was discussed. It was also mentioned that an equivalent method to get a result may be obtained by propagating the one's vector in small increments, 2. This works much like the integrator within a chemical box model, except rather than updating the species concentration with each time step, we move information between each node.

Using this analogy it, therefore, follows that should we reverse the direction of the flow (change the edges from source $\rightarrow$ target to source $\leftarrow$) it would be possible to see where species influence originates, Figure 2.35. As each network is constructed directly from the Jacobian matrix (what is used within the integrator to propagate a model forwards in time), reversing the link direction is analogous to taking the transpose of the jacobian. Within the modelling world, the resultant matrix is now known as the adjoint. The adjoint matrix is often used in the running of models backwards in time, to make historic predictions based on current data.

Some information on using the adjoint and references here



(a) Traditional Influence Graph



(b) Reversed-link (adjoint) influence graph

Figure 2.35: **Link reversal of the Jacobian Sensitivity matrix graph results in a graph of the Adjoint.** Showing how in changing the direction of the links in a graph is equivalent to applying the transpose to an adjacency matrix (right). In the case of a Jacobian based graph, this is analogous to using the adjoint to propagate the model back in time - something that can be used to identify the influence upon a species with a model.

2.0.31 Testing

As with all scientific processes, it is important to first test the algorithm on a small, comprehensible example. To do this we start with the creation of CH_2OO within the Borneo mechanism. This is a direct product of isoprene. In tracing back all the species precursors the mechanism for its creation can be described as:

$$O_3 + C_5H_8 \xrightarrow{\kappa} CH_2OOE + (MACR\ or\ MVK) \tag{2.18}$$

$$CH_2OOE \xrightarrow{\kappa_{\mathrm{dec}}} CH_2OO \tag{2.19}$$

In traversing the adjoint/reversed graph, this presents a singe 'shortest path' between the product and its precursor. This creates a base test for the algorithm. The PageRank algorithm is now run

with a personalisation vector consisting of a value of 1000000 for the species of interest and -1 for all others. A damping factor value of 0.01 is also used for the algorithm.

As CH_2OOE only has one precursor (α-pinene) the initial test is done on this. From this, the identification of isoprene as a source is successful, although since the algorithm is performed on the whole network, there are results for several additional species, Table 2.6. This is because page rank works on using teleportation to change between items in the evolution of the system. With the design of the personalisation vector, these values will, however, be significantly smaller than any containing useful results.

C_5H_8	9.920000e-03
CH_2OOE	9.920000e-01
$C_{816}O$	-9.990000e-07
$NC_{101}CO$	-9.990000e-07
$C_{926}OH$	-9.990000e-07

Table 2.6: A reversed graph Page Rank test with $C_5H_8 + O_3 \longrightarrow CH_2OOE$ as the only reaction.

Next, we apply the same methodology to CH_2OO. This creates the graph in Figure 2.36. Here it is seen that CH_2OO is directly dependant on the radicals $CH_2OO[F, B, C, G, A]$, and CH_2OOE. This is then dependant on Isoprene, which then has a range of dependencies with all have precursors of their own (not shown). Table 2.8 shows the direct dependence on Isoprene and the criegee radicals of CH_2OO in addition to

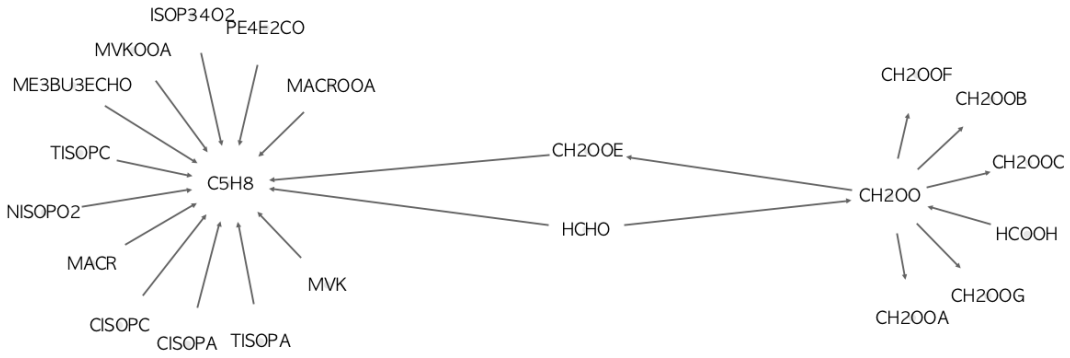


Figure 2.36: **The reversed subgraph between Isoprene, CH_2OOE and CH_2OO.** This is a subgraph of the afforementioned species, showing them and their neighbours. Here the arrows point towards a species precursor.

CH2OO	0.992000
CH2OOE	0.001670
CH2OOF	0.001660
CH2OOG	0.001660
CH2OOA	0.001660
CH2OOC	0.001640
CH2OOB	0.001640
C5H8	0.000016
MACR	0.000016
C2H4	0.000007
HMACR	0.000007
ISOP34NO3	0.000005

Table 2.7: A reversed graph Page Rank test with CH_2OOE, small constent values have been removed.

Next, a test using α pinene with a bit more chemistry is done.

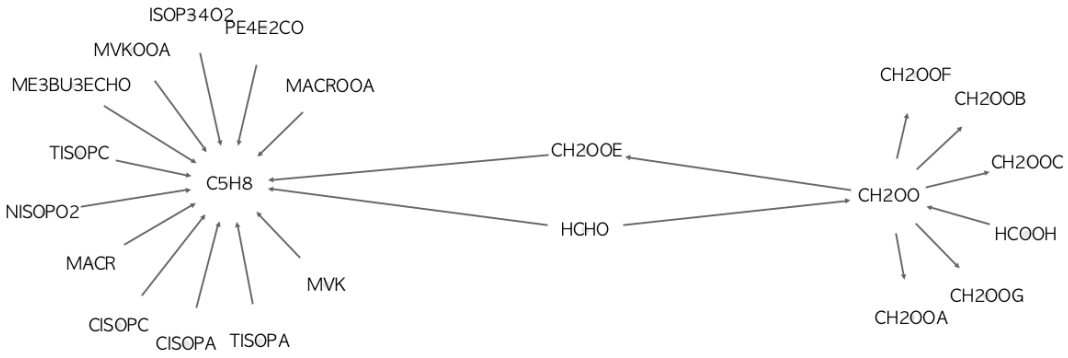


Figure 2.37: **The reversed subgraph between α-pinene, and $NC_{101}CO$** This is a subgraph of the afforementioned species, showing them and their neighbours. Here the arrows point towards a species precursor.

NC101CO	9.920000e-01
APINENE	9.210000e-06
NAPINBO	4.540000e-03
NAPINBO2	2.770000e-03
NAPINBOOH	2.690000e-03
C511OOH	-9.990000e-07
C527NO3	-9.990000e-07

Table 2.8: A reversed graph Page Rank test with $NC_{101}CO$

2.0.32 Source Analysis Using The Jacobian

A bit about the maths, and procedure. This method is much easier and provides more concrete results.

2.0.33 Verdict

As the PageRank algorithm is applied to the whole network and contains teleportation it provides small values for species without a direct link to the species in question. This requires some sort of changepoint analysis to filter. A much simpler method would be the calculation of the shortest simple path between a species in question and all other species, and then subtract the value obtained within each step to get its contribution. for the example A $\xrightarrow{4}$ B $\xrightarrow{6}$ C the shortest path from A to C would be 10 and B to c would be 6. The influence of A on C would be the influence of A on B divided by the total influence on B.

The simplest is to calculate the fraction of A which contributes to B and then multiply the what B contributes to C by that fraction using the jacobian.

This section to be finished when it is not 5 am in the morning.

2.0.34 Conclusions

For large complex graphs, visual analytics may not form a suitable solution. Instead, it is possible to apply a range of mathematical algorithms to tell us what species are important within a network. Chemical mechanisms, much like many real-world graphs, were shown to have both small world and scale-free properties within their structure. This means that they have both a local(social) and global(hierarchical) structure.

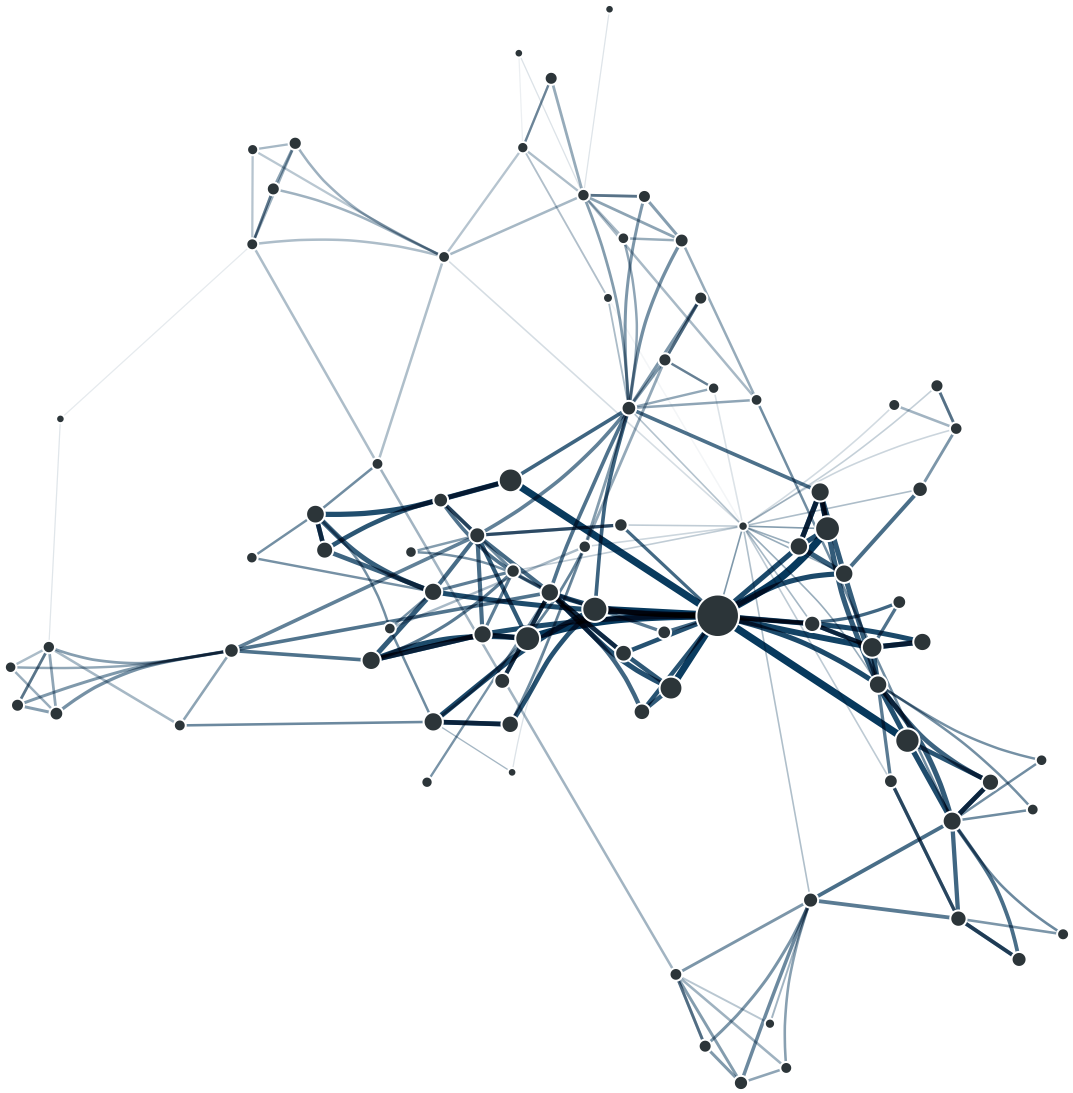
It was shown that it is possible to generate a citation network for papers citing the master chemical mechanism and represent this in the form of a graph. Further exploration into the network structure led to the creation of a co-citation and an author network from the original dataset. This was then used to evaluate several centrality metrics, and their ability to highlight roles within the co-authorship network.

Next, the centrality metrics were applied to a range of chemical mechanisms representing urban, terrestrial and marine environments. Here it was seen that the sum of these follows a similar trend to more traditional methods of evaluating node importance, such as flux and concentration analysis. As this was the case, averaged metric values for each scenario were generated, and the individual chemistry compared with the aid of a TF-IDF algorithm. This highlights important species for each run, whilst ignoring those which are important across all runs.

Finally, it was noted that in reversing the direction of links within a graph it is possible to determine the source of influence on a node. An attempt to do this using the PageRank algorithm was made, although this proved to not be the most effective method to accomplish this. Instead, it was far

simpler to make use of the adjacency matrix (jacobian) and apply the transformation there to get the required results.

In this chapter the merit of using centrality metrics to mathematically analyse a complex network was shown. It is suggested that these are used in conjunction with more traditional methods of simulation evaluation to allow for a greater understanding of the roles each species have within a certain environment.



Bibliography

(2019). Lapack — Linear Algebra Package. http://www.netlib.org/lapack/. http://www.netlib.org/lapack/.

Barabási, A.-L. (2019). Nature-150-Cover.Pdf. *Nature*, 575(7781). https://www.nature.com/immersive/d42859-019-00121-0/public/pdf/nature-150-cover.pdf.

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.

Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2016). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17(4):305–338. https://doi.org/10.1007/s00799-015-0156-0.

Bloss, C., Wagner, V., Jenkin, M. E., Volkamer, R., Bloss, W. J., Lee, J. D., Heard, D. E., Wirtz, K., Martin-Reviejo, M., Rea, G., Wenger, J. C., and Pilling, M. J. (2005). Development of a detailed chemical mechanism (mcmv3.1) for the atmospheric oxidation of aromatic hydrocarbons. *Atmospheric Chemistry and Physics*, 5(3):641–664. https://www.atmos-chem-phys.net/5/641/2005/.

Bloss, W. J., Lee, J. D., Bloss, C., Heard, D. E., Pilling, M. J., Wirtz, K., Martin-Reviejo, M., and Siese, M. (2004). Validation of the calibration of a laser-induced fluorescence instrument for the measurement of oh radicals in the atmosphere. *Atmospheric Chemistry and Physics*, 4(2):571–583. https://www.atmos-chem-phys.net/4/571/2004/.

Bonacich, P. (1987). Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182. http://www.jstor.org/stable/2780000.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social Networks*, 29(4):555 – 564. http://www.sciencedirect.com/science/article/pii/S0378873307000342.

Borgatti, S. P. (2005). Centrality And Network Flow. *Social networks*, 27(1):55–71. http://www.sciencedirect.com/science/article/pii/S0378873304000693.

Boudin, F. (2013). A Comparison Of Centrality Measures For Graph-Based Keyphrase Extraction. https://hal.archives-ouvertes.fr/hal-00850187/document.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177.

Brasseur, G. and Jacob, D. (2017). *Modeling Of Atmospheric Chemistry*. Cambridge University Press. https://books.google.co.uk/books?id=k9_PDgAAQBAJ.

Broido, A. D. and Clauset, A. (2019). Scale-Free Networks Are Rare. *Nature communications*, 10(1):1017. http://dx.doi.org/10.1038/s41467-019-08746-5.

Cabello, R. (2019). Three.Js – Javascript 3D Library. https://threejs.org/.

Cajal, S. R. (2020). Cortex drawings. *web*. https://upload.wikimedia.org/wikipedia/commons/5/5b/Cajal_cortex_drawings.png.

Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-Law Distributions In Empirical Data. *SIAM Review*, 51(4):661–703. https://doi.org/10.1137/070710111.

Cornell, L. (2020). Mark 1 Perceptron. https://en.wikipedia.org/w/index.php?title=Perceptron&oldid=935763442.

de Solla Price, D. J. (1965). Networks of scientific papers. *Science*, 149(3683):510–515. https://science.sciencemag.org/content/149/3683/510.

Derwent, R. G., Jenkin, M. E., Saunders, S. M., and Pilling, M. J. (1998). Photochemical Ozone Creation Potentials For Organic Compounds In Northwest Europe Calculated With A Master Chemical Mechanism. *Atmospheric environment*, 32(14):2429–2441. http://www.sciencedirect.com/science/article/pii/S1352231098000533.

Dillon, T. J., Tucceri, M. E., and Crowley, J. N. (2006). Laser induced fluorescence studies of iodine oxide chemistry part ii. the reactions of io with ch3o2, cf3o2 and o3. *Phys. Chem. Chem. Phys.*, 8:5185–5198. http://dx.doi.org/10.1039/B611116E.

Ebel, H., Mielsch, L.-I., and Bornholdt, S. (2002). Scale-free topology of e-mail networks. *Phys. Rev. E*, 66:035103. https://link.aps.org/doi/10.1103/PhysRevE.66.035103.

Edsu and Ellis, D. (2019). Etudier. https://github.com/wolfiex/etudier.

Ellis, D. (2019). Using Tf-Idf To Form Descriptive Chapter Summaries Via Keyword Extraction. https://towardsdatascience.com/using-tf-idf-to-form-descriptive-chapter-summaries-via-keyword-extraction-4e6fd857d190.

Elshorbany, Y. F., Kleffmann, J., Hofzumahaus, A., Kurtenbach, R., Wiesen, P., Brauers, T., Bohn, B., Dorn, H.-P., Fuchs, H., Holland, F., Rohrer, F., Tillmann, R., Wegener, R., Wahner, A., Kanaya, Y., Yoshino, A., Nishida, S., Kajii, Y., Martinez, M., Kubistin, D., Harder, H., Lelieveld, J., Elste, T., Plass-Dülmer, C., Stange, G., Berresheim, H., and Schurath, U. (2012). Ho X Budgets During Hoxcomp: A Case Study Of Ho X Chemistry Under No X -Limited Conditions. *Journal of geophysical research*, 117(D3). https://www.academia.edu/29719780/HO_x_budgets_during_HOxComp_A_case_study_of_HO_x_chemistry_under_NO_x_-limited_conditions.

Fantin, V., Buttol, P., Pergreffi, R., and Masoni, P. (2012). Life Cycle Assessment Of Italian High Quality Milk Production. A Comparison With An Epd Study. *Journal of cleaner production*, 28:150–159. http://www.sciencedirect.com/science/article/pii/S095965261100388X.

Freeman, L. (1977). A set of measures of centrality based on betweenness. 40:35–41.

Freeman, L., Borgatti, S., and White, D. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13:141–154.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Fujita, M., Inoue, H., and Terano, T. (2017). Searching promising researchers through network centrality measures of co-author networks of technical papers. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 615–618.

Gemma, J. (2019). The Most Influential Men And Women On Twitter 2017. `https://www.brandwatch.com/blog/react-influential-men-and-women-2017/`.

Géron, A. (2017). *Hands-On Machine Learning With Scikit-Learn And Tensorflow: Concepts, Tools, And Techniques To Build Intelligent Systems.* O'Reilly Media. `https://books.google.co.uk/books?id=khpYDgAAQBAJ`.

Goh, K. I., Kahng, B., and Kim, D. (2001). Universal Behavior Of Load Distribution In Scale-Free Networks. *Physical review letters*, 87(27 Pt 1):278701. `http://dx.doi.org/10.1103/PhysRevLett.87.278701`.

Google (2019). Google Scholar. `https://scholar.google.com/schhp?hl=en`.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

Hobson, E. A., Mønster, D., and DeDeo, S. (2018). Strategic Heuristics Underlie Animal Dominance Hierarchies And Provide Evidence Of Group-Level Social Knowledge. `http://arxiv.org/abs/1810.07215`.

Humphries, M. D. and Gurney, K. (2008). Network 'Small-World-Ness': A Quantitative Method For Determining Canonical Network Equivalence. *PloS one*, 3(4):e0002051. `http://dx.doi.org/10.1371/journal.pone.0002051`.

Jacob, P.-M. and Lapkin, A. (2018). Statistics of the network of organic chemistry. *React. Chem. Eng.*, 3:102–118. http://dx.doi.org/10.1039/C7RE00129K.

Jeanningros, Y., Vlaeminck, S. E., Kaldate, A., Verstraete, W., and Graveleau, L. (2010). Fast Start-Up Of A Pilot-Scale Deammonification Sequencing Batch Reactor From An Activated Sludge Inoculum. *Water science and technology: a journal of the International Association on Water Pollution Research*, 61(6):1393–1400. http://dx.doi.org/10.2166/wst.2010.019.

Jenkin, M. E. and Hayman, G. D. (1999). Photochemical Ozone Creation Potentials For Oxygenated Volatile Organic Compounds: Sensitivity To Variations In Kinetic And Mechanistic Parameters. *Atmospheric environment*, 33(8):1275–1293. http://www.sciencedirect.com/science/article/pii/S1352231098002611.

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.

Jenkin, M. E., Saunders, S. M., Wagner, V., and Pilling, M. J. (2003). Protocol for the development of the master chemical mechanism, mcm v3 (part b): Tropospheric degradation of aromatic volatile organic compounds. *Atmospheric Chemistry and Physics*, 3(1):181–193. https://www.atmos-chem-phys.net/3/181/2003/.

Jenkin, M. E., Young, J. C., and Rickard, A. R. (2015). The mcm v3.3.1 degradation scheme for isoprene. *Atmospheric Chemistry and Physics*, 15(20):11433–11459. https://www.atmos-chem-phys.net/15/11433/2015/.

John Hay, Ben Lynch, D. S. (1960). Mark 1 Perceptron Operators' Manual. *Cornell Aeronautical Laboratory*. https://apps.dtic.mil/dtic/tr/fulltext/u2/236965.pdf.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). Scipy: Open source scientific tools for Python. http://www.scipy.org/.

Kleinberg, J. M. (1999). Authoritative Sources In A Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632. http://doi.acm.org/10.1145/324133.324140.

Korke, R., Gatti, M. d. L., Lau, A. L. Y., Lim, J. W. E., Seow, T. K., Chung, M. C. M., and Hu, W.-S. (2004). Large Scale Gene Expression Profiling Of Metabolic Shift Of Mammalian Cells In Culture. *Journal of biotechnology*, 107(1):1–17. http://dx.doi.org/10.1016/j.jbiotec.2003.09.007.

Krebs, V. E. (2002). Mapping Networks Of Terrorist Cells. *Connections*, 24(3):43–52. http://ecsocman.hse.ru/data/517/132/1231/mappingterroristnetworks.pdf.

Kumar, R. and Upfal, E. (2000). The Web As A Graph. http://cs.brown.edu/research/webagent/pods-2000.pdf.

Langville, A. and Meyer, C. (2005). A Survey Of Eigenvector Methods For Web Information Retrieval. *SIAM Review*, 47(1):135–161. https://doi.org/10.1137/S0036144503424786.

Ling, Z. H., Guo, H., Lam, S. H. M., Saunders, S. M., and Wang, T. (2014). Atmospheric photochemical reactivity and ozone production at two sites in hong kong: Application of a master chemical mechanism–photochemical box model. *Journal of Geophysical Research: Atmospheres*, 119(17):10567–10582. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014JD021794.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. https://doi.org/10.1007/BF02478259.

Mohanty, J. G., Nagababu, E., and Rifkind, J. M. (2014). Red Blood Cell Oxidative Stress Impairs Oxygen Delivery And Induces Red Blood Cell Aging. *Frontiers in physiology*, 5:84. http://dx.doi.org/10.3389/fphys.2014.00084.

Molontay, R. and Nagy, M. (2020). Twenty Years Of Network Science: A Bibliographic And Co-Authorship Network Analysis. *arXiv*. http://arxiv.org/abs/2001.09006.

Monastersky, R. and Van Noorden, R. (2019). 150 Years Of Nature: A Data Graphic Charts Our Evolution. *Nature*, 575(7781):22–23. http://dx.doi.org/10.1038/d41586-019-03305-w.

Mquantin (2020). Idf response functions. *wikipedia commons*. https://upload.wikimedia.org/wikipedia/commons/0/05/Plot_IDF_functions.png.

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

Newman, M. E. J. (2004). Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5200–5205. https://www.pnas.org/content/101/suppl_1/5200.

Oliphant, T. (2006). Guide to numpy.

Ottens, A. K., Kobeissy, F. H., Golden, E. C., Zhang, Z., Haskins, W. E., Chen, S.-S., Hayes, R. L., Wang, K. K. W., and Denslow, N. D. (2006). Neuroproteomics In Neurotrauma. *Mass spectrometry reviews*, 25(3):380–408. http://dx.doi.org/10.1002/mas.20073.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. *Stanford InfoLab*, (1999-66). Previous number = SIDL-WP-1999-0120 http://ilpubs.stanford.edu:8090/422/.

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.

Pokroy, B., Epstein, A. K., Persson-Gulda, M. C. M., and Aizenberg, J. (2009). Fabrication Of Bioinspired Actuated Nanostructures With Arbitrary Geometry And Stiffness. *Advanced materials*, 21(4):463–469. http://doi.wiley.com/10.1002/adma.200801432.

poliaktiv (2011). Social Network Analysis: Theory And Applications. `https://www.politaktiv.org/documents/10157/29141/SocNet_TheoryApp.pdf`.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes In C (2Nd Ed.): The Art Of Scientific Computing*. Cambridge University Press, USA.

R. Seeley, J. (1949). The net of reciprocal influence: A problem in treating sociometric data. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 3:234–240.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581–603. `https://doi.org/10.1007/BF02289527`.

Saunders, S. M., Jenkin, M. E., Derwent, R. G., and Pilling, M. J. (2003). Protocol For The Development Of The Master Chemical Mechanism, Mcm V3 (Part A): Tropospheric Degradation Of Non-Aromatic Volatile Organic Compounds. *Atmospheric Chemistry and Physics*, 3(1):161–180. `https://hal.archives-ouvertes.fr/hal-00295229`.

Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269. `https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.4630240406`.

Spencer, R. G. M., Hernes, P. J., Ruf, R., Baker, A., Dyda, R. Y., Stubbins, A., and Six, J. (2010). Temporal Controls On Dissolved Organic Matter And Lignin Biogeochemistry In A Pristine Tropical River, Democratic Republic Of Congo. *Journal of geophysical research*, 115(G3):2069. http://doi.wiley.com/10.1029/2009JG001180.

Stubbins, A., Hubbard, V., Uher, G., Law, C. S., Upstill-Goddard, R. C., Aiken, G. R., and Mopper, K. (2008). Relating Carbon Monoxide Photoproduction To Dissolved Organic Matter Functionality. *Environmental science & technology*, 42(9):3271–3276. `http://dx.doi.org/10.1021/es703014q`.

Turányi, T. and Tomlin, A. S. (2014). *Reduction Of Reaction Mechanisms*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44562-4_7.

Vigna, S. (2016). Spectral Ranking. *Network Science*, 4(4):433–445. https://www.cambridge.org/core/journals/network-science/article/spectral-ranking/99ACDCD0CC1B774AB0041FB16AB43D1B.

Watts, D. J. and Strogatz, S. H. (1998). Collective Dynamics Of 'Small-World' Networks. *Nature*, 393(6684):440–442. http://dx.doi.org/10.1038/30918.

Wood, B. (2014). The Origin Of Humans Is Surprisingly Complicated. *Scientific American*. https://www.scientificamerican.com/article/the-origin-of-humans-is-surprisingly-complicated/.

Chapter 3

Chemical mechanism stratification and analysis using ML and graph clustering.

"Entities should not be multiplied beyond necessity."

- William of Ockham, *Summa Logicae*

3.0.1 Introduction

In the previous chapters, we have discussed visualisation and its role in bridging the gap between data and understanding. We have applied centrality metrics to a chemical network to tell us what species are of importance and experimented in getting machine learning models to learn the chemical structure of the species in a mechanism. This final research chapter provides a brief overview of current mechanism reduction techniques, whilst providing two novel alternatives to aid the process.

Science often deals with the problem of understanding complexity. This may be accomplished through organisation and partitioning, for example, the learning of a new skill through chunking, or the parallelisation of a large mathematical problem. In cases where such methods fail, we are forced to 'disregard' complexity. It is common to approximate an atom as a sphere or the value π as 3 with little consequence to the overall result of a calculation. The process of lumping has long been used to replace a complex, changing process (e.g Quantum Mechanics or Boundary Layer Fluid Dynamics) with a simpler constant process, [Mahajan, 2008]. In such cases, an approximate analysis may be far more useful than a lengthy exact solution, or none at all.

Similar problems of complexity can also be seen within the chemistry of the atmosphere. An example is seen within the Master Chemical Mechanism[1] (MCM), [**?**], this contains 1228 RO_2 reactions. If written explicitly all $RO_2 - RO_2$ interactions would result in a total of 1507984 reactions. Instead, the MCM overcomes this problem by creating a RO_2 pool, with which all RO_2 species react. This results in a mechanism which preserves the quality of science with only 0.000814 of the total possible $RO_2 - RO_2$ reactions.

However even with such simplifications, atmospheric chemical mechanisms have been increasing in size over the last 10 years, **??**REFC1fig. With the ability to automate their construction, mechanisms with species numbers of the millions become possible. Although the existence of more-explicit mechanisms may improve the quality of science produced, they can cause problems for efficient computation, diagnosis and analysis. This chapter shall look at two methods in which we may simplify a mechanism by grouping similar species together. These are through the use of species lifetime (Subsection 3.0.18) and graph-based clustering (Subsection 3.0.10).

3.0.2 Mechanism Reduction

As discussed, the first step to simplifying a complex task involves the partitioning data into categories. For a mechanism, we begin by looking at the reaction or species which are related to the area that is

[1] Version 3.3.1 .

being researched. Items are partitioned into important, needed and redundant categories (described below).

- **Important** - reactions or species directly related to the topic / outcome we are interested in
- **Needed** - reactions/species required by the important species, such that they may perform their desired function
- **Redundant** - those we may remove with little or no consequence to the final outcome of the model.

3.0.3 Reaction Removal

Since atmospheric chemical mechanism forms a numerically stiff system, a reduction in the number of reactions within a mechanism leads to a reduction in the computational burden experienced by a model each iteration forwards in time. Classically the identification of important reactions may be accomplished through the use or rate of production and loss analysis (SEC REF). This allows us to filter reactions contributing less than 5% to the formation of any species we are interested in. Other methods using principal component analysis of the sensitivity of species (PCAS) also exist and are discussed in Vajda et al. [1985].

3.0.4 Species Removal

Similar to reaction removal, the removal of species is useful because the removing or combining of species inherently reduces or simplifies the reactions within a mechanism. This method also has added benefit of reducing the size of the jacobian matrix used to propagate the chemical system forwards. For large systems which do not use a sparse framework, storing a n^2 matrix in memory can prove difficult.

Many methods of species reduction are possible. The simplest of these is through the use of trial and error[2] [Turiinyi, 1990] (Method 1). Here the consuming reactions for a species are removed, and if the resulting deviation in results between the full and reduced mechanism is small, their results are kept. The main downside to this is that it only works on a per-species level, which may be very resource-consuming for large mechanisms.

With the use of sensitivity analysis, it is possible to remove species whose reaction are much slower than the rate-determining steps of a mechanism, [Oran and Boris, 1991]. However, even after removing all slow-reacting species, those on a fast timescale remain. Here the use of Quasi-Steady-State

[2] A tried and tested method for scientific discovery.

Approximation (QSSA), [Whitehouse et al., 2004a], can be used to identify species associated with fast timescale reactions. QSSA works on the assumption that such species have little to no change in concentration over time - i.e. the net flux ($\dot{v_i}$) is zero. Such an assumption causes an error Δc_i of :

$$\Delta c_i = \frac{\dot{v_i}}{J_{ii}} \tag{3.1}$$

where J_{ii} is the diagonal of the Jacobian matrix. Here if the error for a species is small, the species may be removed from the mechanism.

Finally, investigation of the system Jacobian can be used to identify redundant species, which is a 'capable' and 'efficient' method for removing most redundant reactions and species from the MCM, [Whitehouse et al., 2004a]. Use of a log-normalised Jacobian to determine which species can be removed is found in the connectivity method Turányi and Tomlin [2014]; Turányi [1990]. Here the influence a 1% change in a species concentration has on the concentration of 'important' species can be determined by

$$B_i = \sum_j ((y_i/f_i)(\partial f_i/\partial y_i))^2 \tag{3.2}$$

where $(y_i/f_i)(\partial f_i/\partial y_i)$ is element i of the normalised Jacobian. Through an iterative process species with a low contribution to our important species can be found and removed.

3.0.5 Lumping

Rather than removing species or reactions from a mechanism we may combine them to form a new composite species. This is species lumping. To do this we must first consider how we determine species that are to be joined together, and then how their grouped reactions will contribute to every other species it reacts with. Some of the more general types of lumping styles are outlined below.

3.0.5.1 Chemical Lumping

Mechanisms follow protocols in their generation. This produces reaction styles that many like-structured species follow in their degradation. In determining such classes we may be able to generalise like-species reactions and group them as one. An example of this is the Common Representative Intermediates (CRI) Mechanism (described in **??**). Here the ozone production potential of the species within the MCM is used to simplify and reduce it. Species with a similar C−C and C−H ratio (their CRI index) are lumped into a single representative species. Alternatively, time scale analysis for species lumping has been successfully applied by Whitehouse et al. [2004b]. Here it is seen that many

groups of species have coefficients that are identical or sufficiently similar, resulting based on their type. This results in a similar overall lifetime for species in the same group, allowing them to be lumped together with little overall consequence to the final result of the simulation.

3.0.6 Data Setup

Unlike manual reduction, this chapter does not concern itself with the intricacies of the chemistry behind a mechanism. Instead, we search for an automated method of simplifying the mathematical structure behind a mechanism whilst preserving the quality of science it represents. Although this may not directly replicate real-world scenarios, it can provide an accurate test of the robustness of a mechanism and the equations within it. I work on the assumption that the equations describing each reaction are representative of experimental results, and in simplifying these, their usefulness in modelling the real data is preserved. This section describes the experimental setup for the experiment.

3.0.7 The Mechanism

The mechanism used is the Common Representative Intermediates (CRI) Mechanism v2.2,[Jenkin, 2019]. This is an already reduced version of the MCM, where species are grouped based on their ozone formation potential - i.e. the C−C and C−H ratio of bonds. Reductions have been made on a compound-by-compound basis and compared to the MCM using a series of 5-day box-model simulations, [Jenkin et al., 2008].

Why further simplify the CRI network?

CRI v2.2 is a mechanism of 422 species and 1261 reactions. Although this is significantly smaller than the full MCM, it may still prove problematic if used within a global model - for comparison the GEOS-Chem[3] standard chemistry is approximately half the size of this, Community [2020].

3.0.8 The Box-Model

The box model used shall be an adapted version of the Dynamically Simple Model of Atmospheric Chemical Complexity (DSMACC) Emmerson and Evans [2009]; **?**. This has had several changes which allow for multiple parallel runs, easy extraction of rates, fluxes and the jacobian matrix as well as a simple Ncurses interface for loading and parsing new files.

[3] A global 3D model of atmospheric chemistry driven by meteorology from NASA's Goddard Earth Observing System (GEOS), [GEOS-Chem, 2020].

The DSMACC model works by using the Kinetic PreProcessor (KPP), [**?**], to generate Fortran code, which can then be used to integrate the provided mechanism. As there were some issues presented with this a pre-pre parser code was used on the mechanism before running KPP, and a post parser on some of the files to provide the desired output.

3.0.9 Model Inputs

The aim of this experiment is not to replicate a specific case study or scenario. Instead, we extract all non-lumped species which appear in both CRI and the MCM and provide an assortment of initial condition concentrations to cover the entirety of the input space.

To select the initial conditions there exist several sampling styles Mckay et al. [2000]. The most common style is the random or 'Monte Carlo' approach, however, this does not guarantee a homogeneous distribution of points. A lattice or grid approach is also possible, but that can result in a large number of sample points to produce a complete distribution of the input space. To overcome this a Latin hypercube can be used. This is a generalisation of the Latin square - a square matrix containing n items, arranged in such a way that they only appear once in each row and column (akin to a sudoku puzzle) Dodge [2008]. The experimental setup uses a Latin hypercube to define the initial condition range for 148 species and 300 simulations follow the formula below:

$$\text{concentration} \begin{cases} min = 10^{-8}\ max = 10^{-13}, & \textbf{if} NO, NO_2, O_3 \\ min = 10^{-8}\ max = 10^{-13}, & \text{otherwise} \end{cases} \tag{3.3}$$

3.0.10 Graph based reduction

It has been shown that the graph-based representation of the atmospheric chemical network proves useful in both the visual and mathematical analysis of simulation results (**??**). It, therefore, follows that the network representation of mechanism may also have its uses in the simplification, and thus reduction, of chemical complexity. This section will outline the basic methods of modularity (cluster) detection with the graph framework, the different methods in which this may be done and eventually apply it to a case example representative of the chemistry within the London environment.

3.0.11 Graph Parallels.

EDIT

Although many graph-based methods exist within the reduction realm, most of these concentrate on

the generation of skeletal methods through the building of a directed tree (a subcategory of graphs from source to target) - LIST of refs and sentence of all skeletal methods. Path flux analysis (Sun et al 2010)

Instead, we may find ourselves applying graph theory to solve other reduction methods. For instance, we can trace back influence through connecting edges using Dijkstra's shortest path algorithm (CH2 ref) - analogous to the connectivity method, or a leave one out approach combined with PageRank to access the effects of removing a node.

Graph structure can be used to analyse changes of reactions or relationships between species - providing an alternative representation and method to access such data. Additionally, clustering techniques may be used to locate groups of highly connected, fast reacting/strongly related species. This has applications in both understanding the data, but more importantly chemical lumping. In creating a graph from a model simulation, we not only encode information about the chemical structure, but also the influence between species in the mechanism. By grouping species which have a strong dependence upon each other, we can simplify the provided network or mechanism.

3.0.12 Types Of Graph Clustering

Unlike vector clustering algorithms (such as DBSCAN, UMAP and K-means - see **??**), graph clustering metrics do not rely on the spatial orientation of the data to determine groups or 'clusters'. Instead, these may partition the network into segments, group nodes by structural equivalence or explore the 'flow' dynamics of the network.

Algorithms such as Label Propagation [Raghavan et al., 2007] and spin-glass [Newman and Girvan, 2004] work by randomly assigning nodes with property or label. This property is then transferred to its neighbours. Other algorithms such as the nested block model can decompose a graph into clusters of like properties, [Fortunato, 2010]. These are often grouped in the form of topological equivalence which can be either:

- *structual equivalence* - vertices are similar if they have like neighbours, [Zhou, 2003].
- *regular eqivalence* - retrieves nodes with similar connection patterns (e.g. parent - child node hierarchicl structures), [Everett and Borgatti, 1994].

This works in a similar way to an autoencoder (ref auto **??**), where topological similarities are used to simplify (or encode) the network structure, in a way which it may be again decoded.

Finally, there exist a set of 'flow' based models which use the network dynamics to determine the modularity of a network. These are discussed below.

3.0.13 Walk/Flow-Based Clustering

Temporal networks result in a change in the relationships between items (magnitude/type). Such changes in the network dynamics are encoded within the edges of a graph. To account for this, the primary function of a random walk or 'flow' algorithms is to capture the changes between the real-world systems represented by the network.

In (SECTION SILHO) the silhouette coefficient was used to compares the vector position of clusters with regards to the distance of data points between them. Translating this to the graph framework, topological (graph) clustering defines a cluster, or module, as a region with a greater inter-cluster degree or density[4] compared to their intra-cluster density[5]. This results in a system, that is sorted by group, has more links between elements of the same group than with those in other groups - such patterns can be seen within the sorted adjacency matrix in (Chapter 1 REF).

Since flow-based methods are more interested in the network dynamics, than structure, the number of links or density is replaced with the time a random 'walker' spends 'trapped' between a set of nodes. A real-world analogy would be to view the flow of water in a slowly filling river, Figure 3.1. Here a walker (or water molecule) traverses the entirety of the river/graph network, occasionally getting trapped between a set of nodes. Here although the water is still moving, it ends up spending more time going back and forth between a set of nodes, than exploring the rest of the network. It is these regions of stalled progress that form network clusters.

[4]The number of links or edges between items in the same group.

[5]The number of edges to other clusters

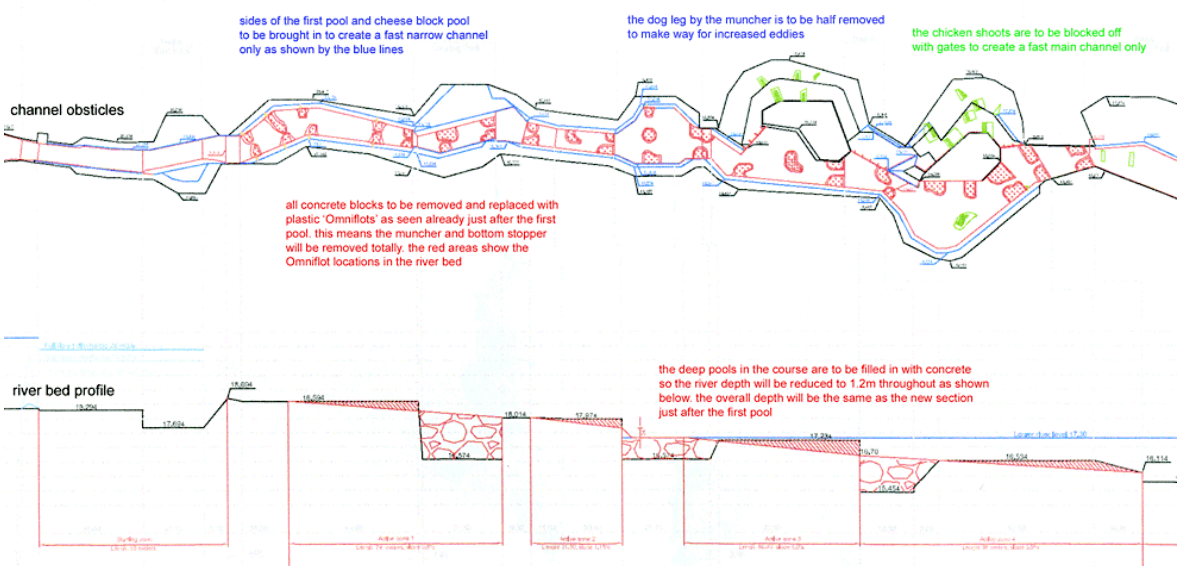


Figure 3.1: **The proposed plans for the change of the UK National Watersports Centre Whitewater Course (Holme Pierrepont).** Walk based clustering is analogous to the movement of a river. Clusters (or modules) are identified as areas where the 'flow' becomes trapped, much like water in the pools immediately following a hydraulic jump. Source: Cornes [2008]

3.0.14 Louvain Clustering

The Louvain clustering algorithm is one of the most popular of the clustering algorithms due to its algorithmic and qualitative robustness, [Blondel et al., 2008; Lu et al., 2015]. On the simplest level, this works by maximising the modularity for each configuration. Modularity is a value between positive and negative unity which measures the density of edge between inter and intra communities and compares it to an equivalent random network. The Louvain is a hierarchical clustering algorithm, this means that after each iteration all nodes which belong to the same cluster are consolidated to form a new 'grouped' item. Inter-cluster links are converted into self-links, and intra-cluster links are updated accordingly [REF INCLUDE LAYERS OF hierarchy VCRI'

3.0.15 Infomap For Graphical Clustering

Similar to the Louvain algorithm is the Rosvall and Bergstrom [2008]'s Infomap. Here each node within the network is assigned its module. These are then perturbed to neighbouring nodes should such a mode lead to a decrease in the map equation (a flow-based method which operates on system dynamics rather than structure - [Rosvall et al., 2009]). The process is repeated until no further reductions are possible.

TWO LEVEL

MULITLEVERL

3.0.16 Selection Criteria For Graph Clustering

The main two criteria in selecting an algorithm for grouping atmospheric reactions are:

1. The algorithm can deal with a directed network - chemistry is directional.
2. The algorithm can handle temporal data - The chemistry within a system changes depending on the time of day. This is mainly due to the change in the amount of sunlight available to photolysis reactions.

As the InfoMap algorithm implements a directed approach coupled with a multi-level clustering approach able to capture node-layer interaction in temporal networks,[Aslak et al., 2018], it makes a good candidate for the task of mechanism reduction.

3.0.17 Evaluation Of InfoMap On A Real Simulation.

Using the initial conditions for London from (Table 2.4), a spun up simulation run with the CRI v2.2 mechanism was run. Since this does not contain $C_5H_{11}CHO$, MVK, MACR or Limonene, these species are omitted from the initialisation. Following a spinup to steady-state, a graph is generated for noon after 1 day of an unconstrained run. The InfoMap algorithm is then applied to the generated graph.

The coarsest level of clustering is shown in Figure 3.2. Here nodes are coloured by their cluster, and approximate polygon hulls surround the nodes closest to the median cluster centre. Much like the findings in (CHAPTER STRT), it is seen that different sections of the graph network represent different types of chemistry - for example, hull 4 contains aromatic species, hull 2 contains the products of linear alkanes and hull 3 contains the terpenes.

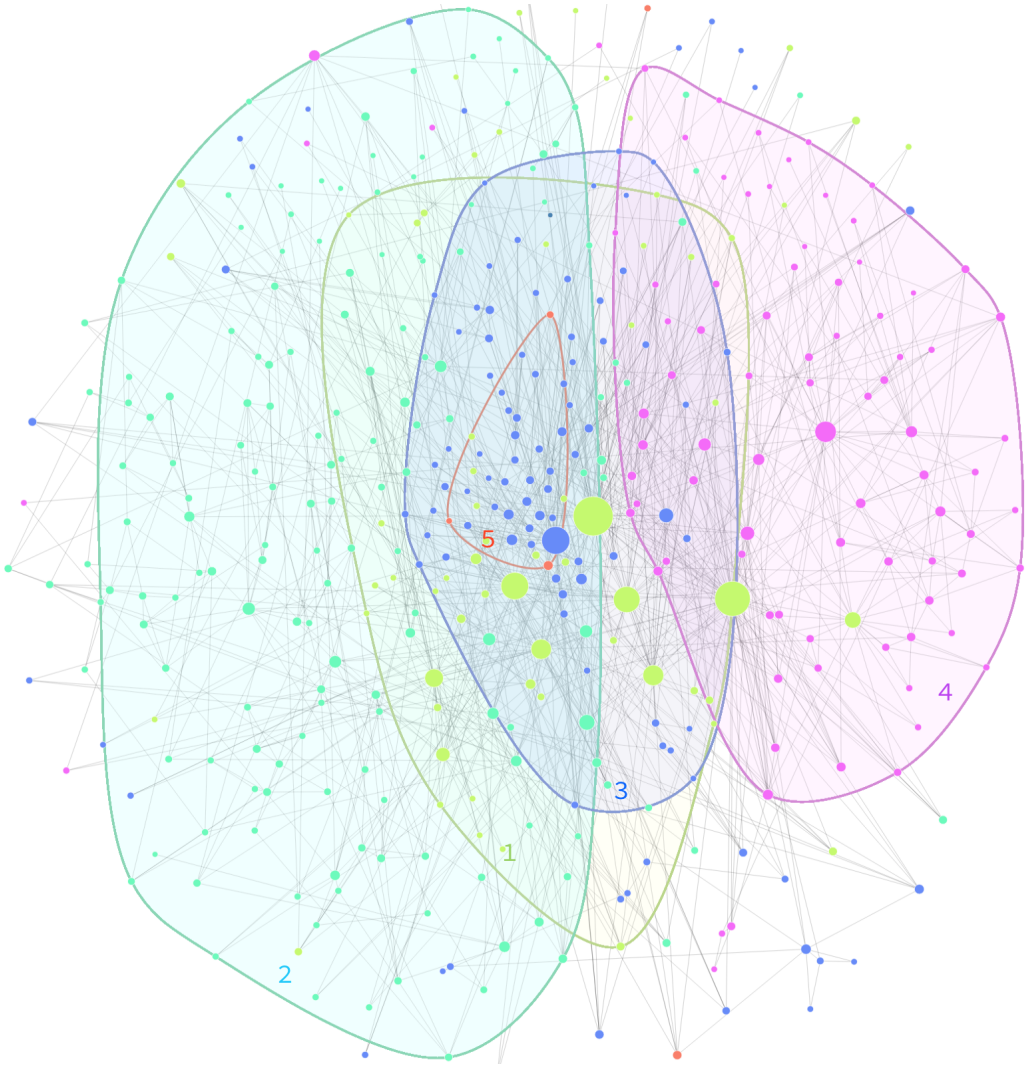


Figure 3.2: **A graph of CRI v2.2 showing the hulls of the first level of hierarchical clustering.** Nodes are coloured by the splits in branches, and the hulls enclose the nodes which lie within 95% of the (median) centre of the cluster.

Since the InfoMap provides a finer level of clustering which has originated from this, it is important to evaluate this. Using a graph-hull approach, as in Figure 3.2, becomes cluttered and unusable. Instead, a bubble plot may be used. Although this sacrifices the ability to view links, it allows for the complete overview of the hierarchical structure. In Figure 3.3 shows the nested structure of each clustered group. In an electronic mail correspondence with **?** the origin of the naming convention of reduced species was explained. Using this, individual nodes are coloured by their prefixes. This allows the further categorisation of the species structure within each category.

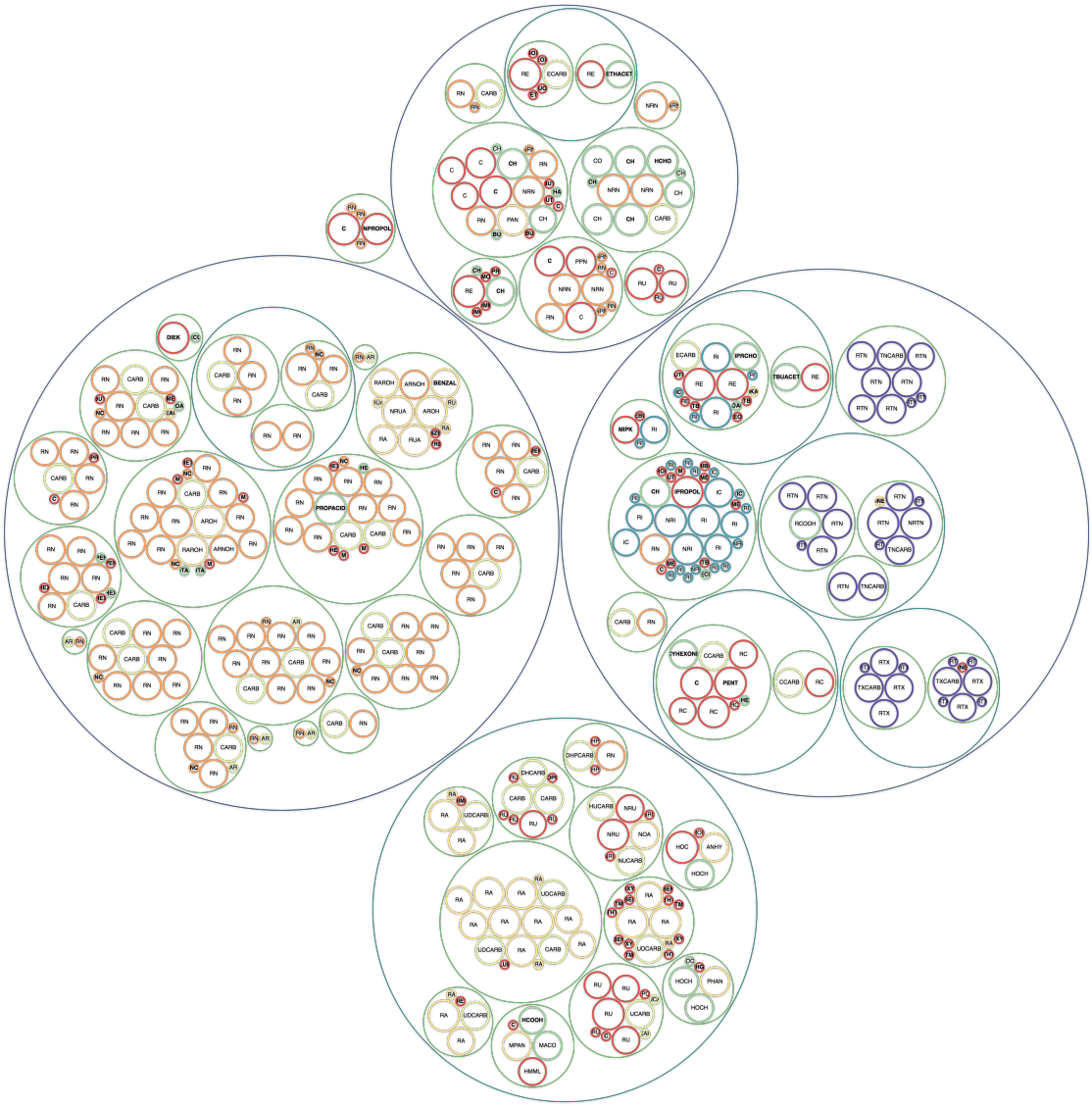


Figure 3.3: **Species structure within each cluster.** A nested bubble chart is used to show the full hierarchical structure of the mechanism. This allows us to evaluate the species structure/type that has been extracted in each level of the hierarchical split. Node sizes are representative of the $\log_{10}$ number of walkers that have become trapped by the flow algorithm at a location.

3.0.17.1 Species Type And Clustering

Although the nested bubble chart is an intuitive way to represent groups within a graph, a tree approach is more suited to revealing the hierarchical structure of the network, Figure 3.4. Here branches are numerically labelled on each level, allowing us to navigate the structure using a sequence of numbers (e.g. to get to 1.5.C_4H_6 we take the first branch from the centre, followed by the fifth branch after that).

This split notation allows a general overview of the mechanism structure, as well as the reasoning/process of the clustering algorithm. The first level split in Figure 3.2 shows branches 1,2 and 5 to have origins in the linear (n-) alkane species. This can be seen through both the emitted species (bold)

and the *RN* prefix of the species. Here the linear alkanes can react with OH to extract hydrogen and then from a RO_2, or produce a carbonyl *CARBxx*, which can then go on to produce the $RNxxO_2$ peroxy radical.

Except for benzine in 2.14, branches 3 and 4 contain the aromatic species in the network. Branches 4.{2,5,9,11} all consist of $RAxxO_2$ species, which are the product of the addition of OH to toluene/benzine ringed species. 4.{1,7,8} and 1.5 contain peroxy radicals formed from the degradation of conjugated dienes (two alkene groups separated by a single bond, where some sharing of electrons may occur) $RUxxO_2$. For the CRI v2.2 mechanism these are only isoprene and 1,3-butadiene. Such peroxy radicals often go on to form unsaturated carbonyls, as denoted by *UCARBxx*.

Branch 3 contains the monoterpenes. This can be seen in 3.{2,5} ($\alpha-$pinene) and 3.6 ($\beta-$pinenen). Here peroxy radicals formed from the reaction with the e**n**docyclinc[6] and e**x**docyclinc[7] double bonds of $\alpha-$ and $\beta-$ pinene are denoted with the prefix *RTN* and *RTX*.

The $RIxxO_2$ prefix was originally used for the peroxy radicals iso ('i-') alkanes and their carbonyl products - branches 3.{1,4}, however, they tend to mainly be used for smaller branched precursors which produce acetone (CH_3COCH_3) as a major product in their oxidation chain (branch 3.1). As acetone is a particularly unreactive carbonyl, the fact that it is water-soluble means that they may be washed out of the atmosphere by precipitation, [Andersson-Sköld et al., 1992]. This may have been seen to interrupt the ozone formation process under regional-scale photochemical smog conditions in north-western Europe [- from M.Jenkins PAPER? do you know what this is].

Finally, since the CRI index is representative of the oxidation potential it is common to see species containing the CRI value within a cluster. Cluster typically contain a combination of carbonyl ($R(=O)R'$,*CARBxx*), hydroperoxy ($R-OOH$,*RxxOOH*), peroxy ($ROO\cdot$, $RxxO_2$) and nitrate ($R-ONO_2$,NO_3) groups. For the lumped species, it can be common for a RO_2 species to react with NO or NO_3 to produce a carbonyl with a CRI index of two values lower. This can be attributed to the loss of oxygen and the formation of a double bond? (what is the long reaction for the MCM, it seems less direct). Similarly, a reaction with NO or HO_2 can produce a hydroperoxy or nitrate species, which in turn react with OH to produce the an equivalent carbonyl.

3.0.17.2 Number Of Clusters

Sometimes it may be required to have preset (target) number of clusters. The InfoMap algorithm contains a *preffered number of modules* parameter which can either terminate the algorithm early, should the number be reached (or continue splitting if it has not). Since we are interested in merging

[6] Inside the pinene ring.

[7] Outside the pinene ring.

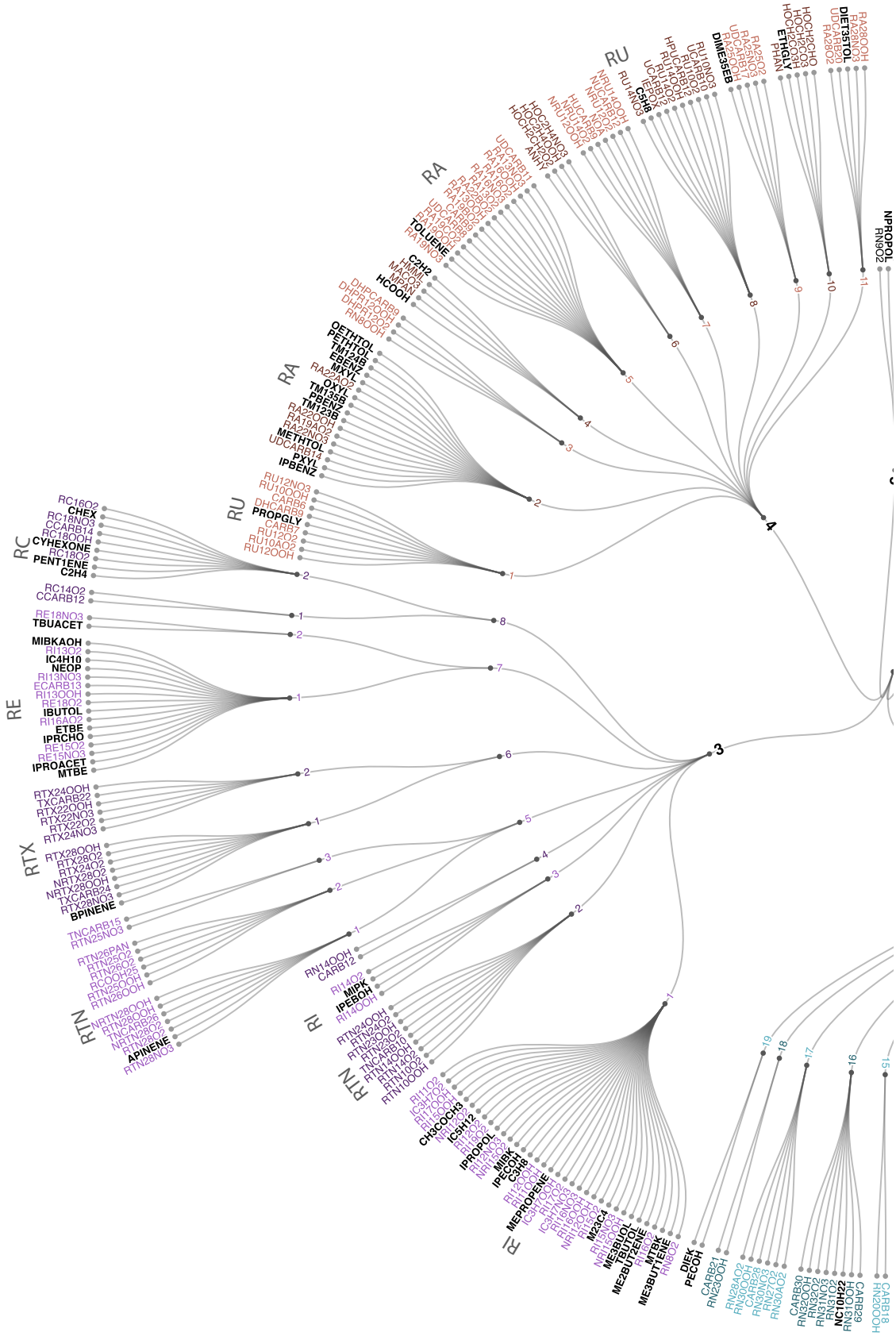
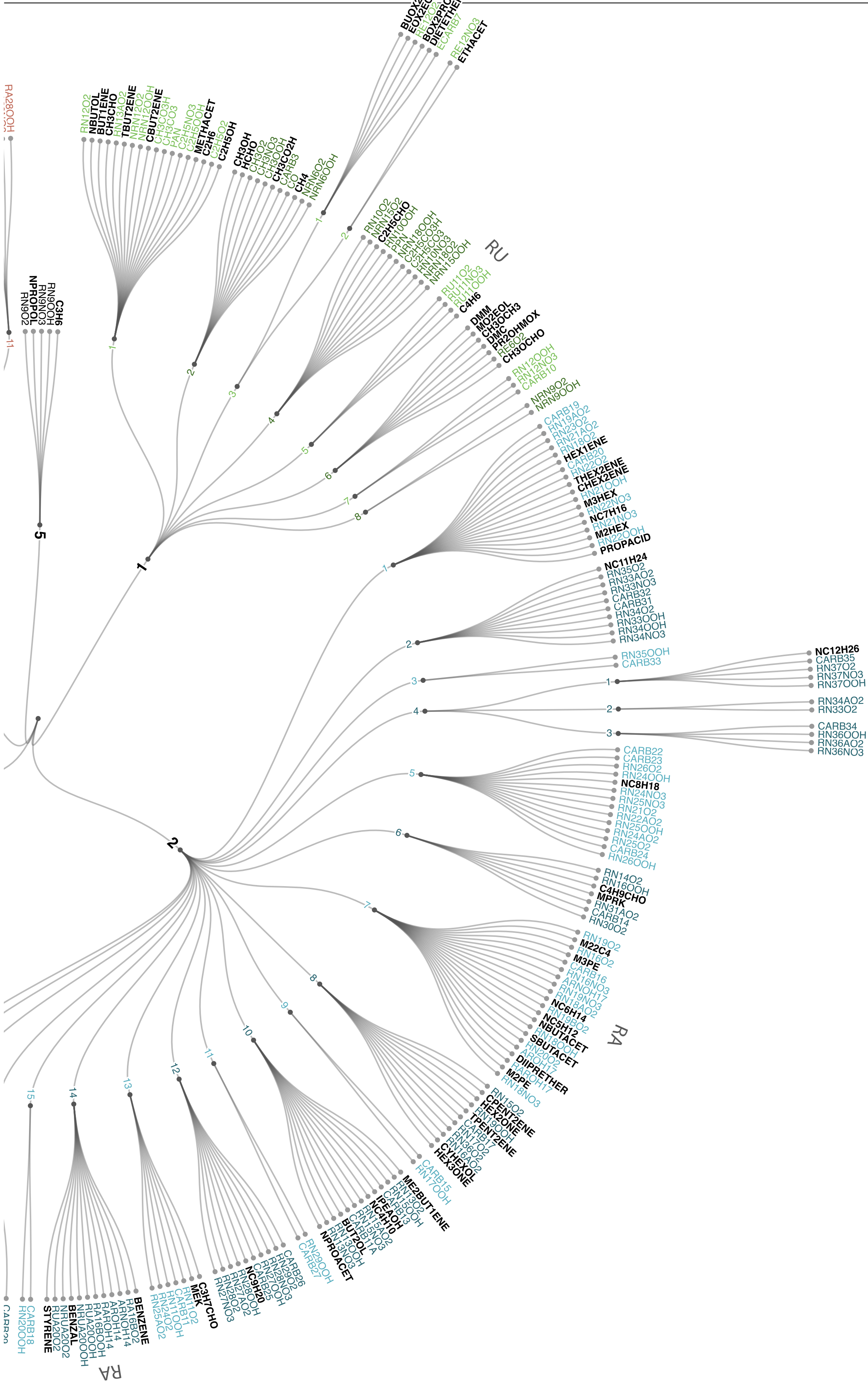


Figure 3.4: **A radial treemap showing the hierarchical clustering of the CRI mechanism.** The simulation results used are representative of the chemistry within London at Noon local time and generated using DSMACC and the InfoMap algorithm.



RU
RA
RA
1
2
5
NC12H26
CARB35
RN37O2
RN37NO3
RN37OOH
RN34AO2
RN33O2
CARB34
RN36OOH
RN36AO2
RN36NO3
RN35OOH
CARB33
NC11H24
PROPACID
NC8H18
C3H6
NPROPOL
ETHACET
BENZENE
STYRENE

smaller numbers of nodes, this can be seen as a useful parameter to have. However, in selecting a number too large, (e.g. 200 clusters, which should result in groups of 2-3 nodes), it is seen that much of the hierarchical information from the network is lost, Figure 3.5. It is for this reason that forcing the number of nodes without reason will not be attempted.

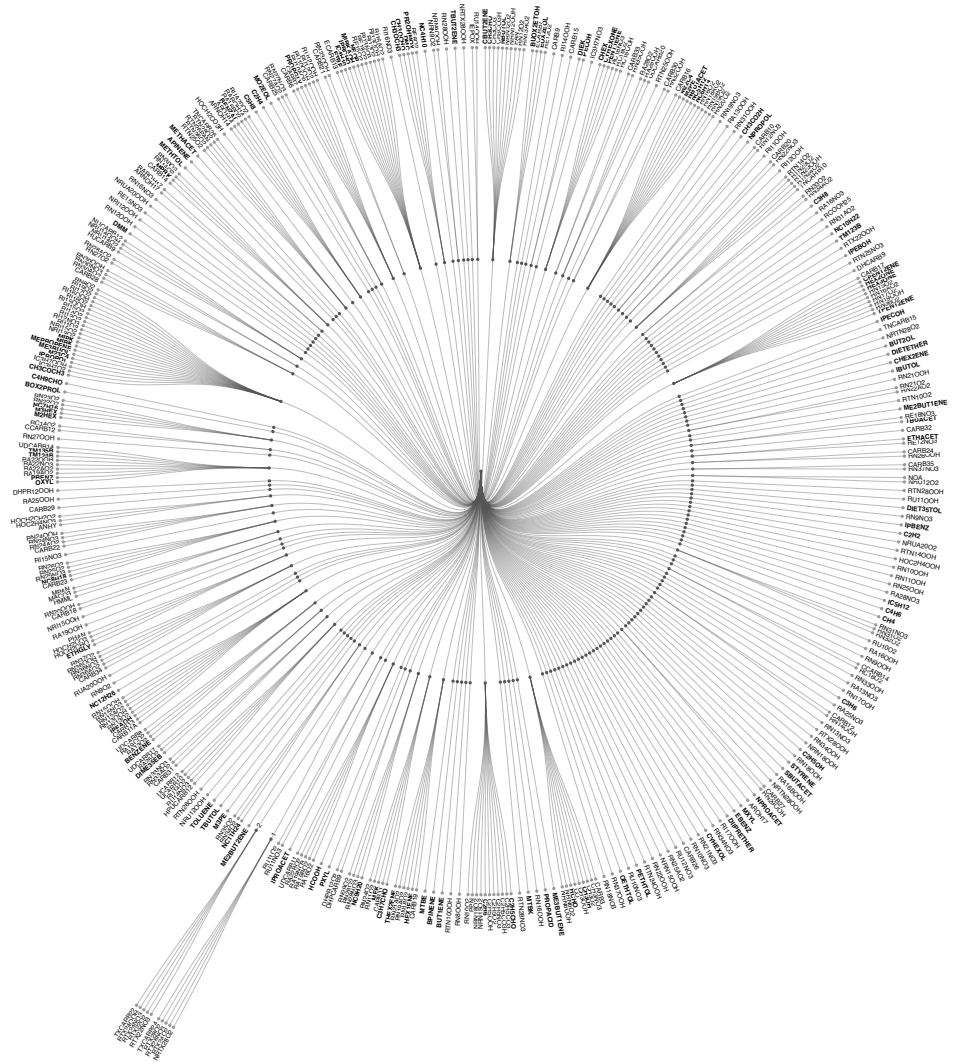


Figure 3.5: **A radial tree of the InfoMap algorithm with a forced number of groups.** Here a loss of hierarchical structure can be seen when compared to Figure 3.4. By setting a high number of required clusters, many species are grouped by themselves, which does not provide a useful output for mechanism lumping.

3.0.18 Reduction through Lifetime

In Subsubsection 3.0.5.1 is was mentioned that a species lifetime could be used to decide on which species may be lumped together. Here Whitehouse et al. [2004b] found that large groups of species within the MCM had similar or identical lifetimes and that in many cases this could be attributed to similar/identical rate coefficients for the same type of reaction. This was then used as a methodology for automatic mechanism reduction.

Using such an approach this section describes a method in which this may be performed without prior knowledge of the mechanism. Natural language processing tools are applied to first determine species of a similar lifetime across a range of timesteps (Subsubsection 3.0.19.1) and then their standardised temporal profiles are compared (Subsubsection 3.0.19.2). However, we first begin by defining the lifetime of a species.

3.0.18.1 Calculating The Lifetime

Lifetime is often defined as the time it takes a quantity to halve. In chemical simulations, this translates as the time it would take for a species concentration to decrease by two for a case where the production flux of that species is 0, and all rate constants remain constant. For the first-order decay of sample Equation 3.4, we can represent the decay using Equation 3.0.18.1. This shows that the half-life of a species is independent of initial concentration.

$$A \rightarrow [k]B \tag{3.4}$$

$$s(t) = a_0 \exp(-kt)\frac{a(t)}{a_0} = \exp(-kt)$$

linearised this gives

$$\ln(\frac{a(t)}{a_0}) = -kt$$

after $\tau_{1/2}$ the concentration is equal to $a_0/2$ of initial rate a_0, which gives

$$\ln(\frac{\frac{a_0}{2}}{a_0}) = \ln(\frac{1}{2}) = \ln(2^{-1}) = -\ln 2 = k\tau_{1/2}$$

$$\tau_{\frac{1}{2}} = \frac{\ln 2}{k} \tag{3.5}$$

In species of the first order only, this may simplified to

$$a(t) = a_0 \exp(t\sum_j k_j)$$

and therefore the half life may be written as the reciprocal sum of rate coefficients:

$$\tau_A = 1/\sum_j k_j \tag{3.6}$$

and is how lifetime is calculated for photochemical species [ref! modelling book, which references

pilling and seakings]. An alternative method for half life calculation may be obtained using the diagonal (self reference) of a Jacobian matrix ,[Turanyi and Tomlin, 2015; Whitehouse et al., 2004b]:

$$\tau_1 = -\frac{1}{J_{ii}} \tag{3.7}$$

This value J_{ii} will usually be negative unless a species does not contain a consuming reaction, then it will be zero.

3.0.19 Comparing Magnitude And Direction

In a simulation the production and loss of species are often directly, or indirectly by other species, influenced by sunlight. This means that the overall production loss fluxes will change with the azimuthal angle of the sun and by consequence the time of the day. As lifetime is calculated using the loss flux of a species, a method that can take temporal changes into account is required to perform lifetime analysis. To do this, a vector containing how a species lifetime changes over the course of a simulation may be obtained from the simulation Jacobians.

These vectors can then be compared by calculating the euclidean (magnitude) and cosine (angle) distance between pairs of species.

3.0.19.1 Euclidian Distance

This is the simplest method of vector comparison and works by calculating the distance between all points in two vectors. For the vectors

$$v1 = [a, b, c, \dots n]$$

$$v2 = [i, j, k, \dots z] \tag{3.8}$$

This can be done using pythoagoras' theorem in Equation 3.9:

$$e_{dist} = \sqrt{(a-i)^2 + (b-j)^2 + (c-k)^2 + \cdots + (n-z)^2} \tag{3.9}$$

This transformation converts the straight line distance between each vector into metric space, allowing us to represent the difference in their magnitudes as a single scalar. Unfortunately, as this requires the difference between all permutations of rows, it cannot be done as a single operation, but as multiple.

3.0.19.2 Cosine Distance

Similarly, if we wish to calculate the angle between two vectors we may use the cosine difference. In starting with the definition of the dot product

$$v1 \cdot v2 = \|v1\|\|v2\| \cos\theta$$

this may be arranged

$$\cos\theta = \frac{v1 \cdot v2}{\|v1\|\|v2\|} \tag{3.10}$$

The problem is that for a meaningful representation for the cosine inequality, the Cauchy-Schwarz (triangle) inequality needs to be satisfied. This states that for all sequences of real numbers a_i and b_i:

$$(\Sigma a_i^2)(\Sigma b_i^2) \geq (\Sigma a_i b_i)^2 \tag{3.11}$$

To account for this each vector needs to be normalised before the calculation of the angle. Although this eliminates any information about the magnitude of the vectors it allows a better comparison of the distribution (or shape) of the initial vector. This normalisation factor makes it particularly useful in the analysis of text documents, where a word may appear multiple times in different length segments.

3.0.20 Temporal Lifetime Vector Comparison

To compare a species diurnal profile with its absolute lifetime we can plot the cosine and euclidian distance against each other on a $x - y$ scatterplot, Figure 3.7b. In this subsection, we compute the Euclidean and cosine distances for all remaining reaction pairs (88410 pairs) for a single simulation. We start by looking at the species density profiles, Figure 3.6.

NOTE: the kernel density plot x axis shows 1-the value shown in the scatter plot. This is because output values from each similarity closer to 1 are more similar. In the scatter plot inverting this however proved simpler to plot and explain

Similar to Whitehouse et al. [2004b], we find there are several groups of species with similar lifetimes. In general, we have two main peaks where the temporal profile and concentration differences are similar. Here the first peak (Figure 3.6 from the right) shows a large agreement between both similarities. This suggests most of the species within this section react similarly, and will very likely have the same inorganic reactions at a similar rate. The second peak, however, shows species which have a similar diurnal response, with different magnitude differences. These species are likely affected by photolysis reactions directly or indirectly but have a differing set of reactions controlling them. In

a concentration line-plot we would expect them to have peaks in the same location, but to change at a different rate/magnitude to each other.

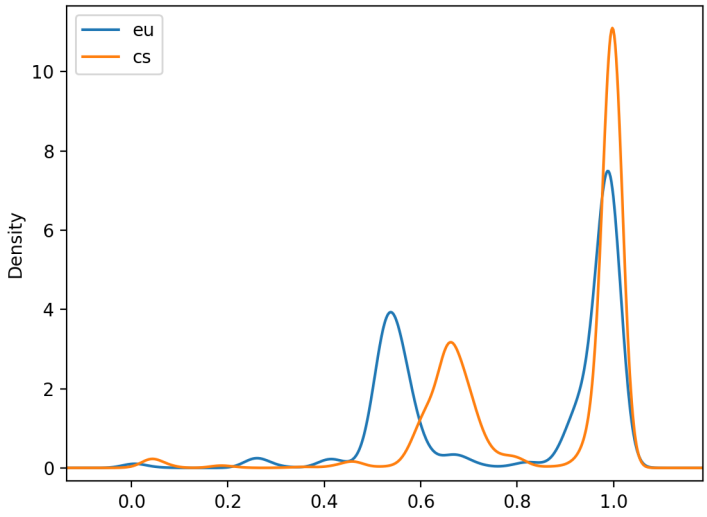
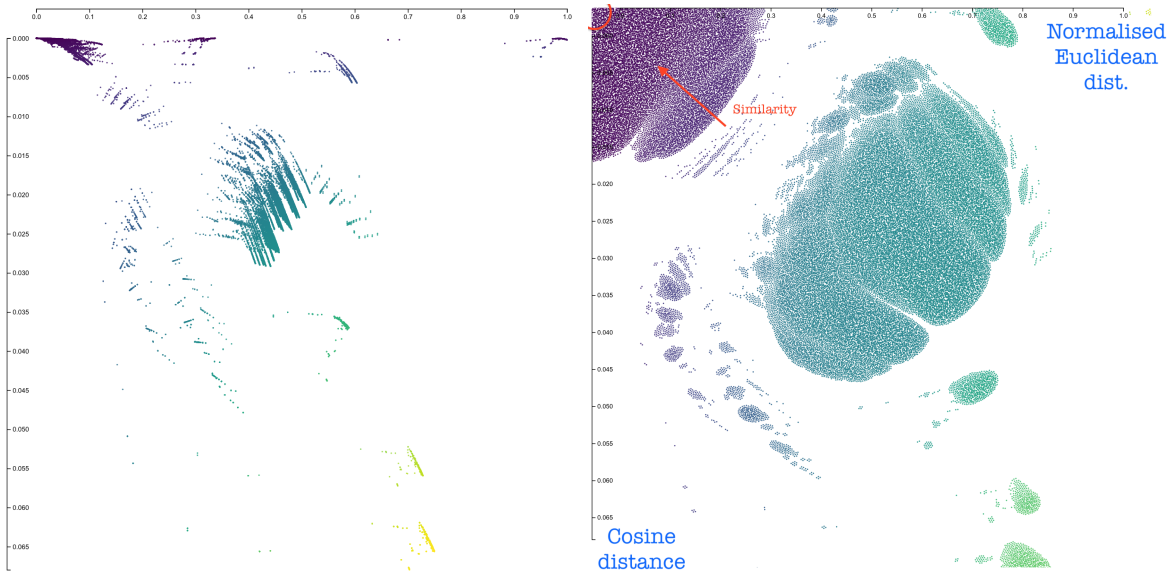


Figure 3.6: Gaussian Kernel Density Estimate plot showing the distributions present for the {0,1} scaled euclidean and cosine distances.

A comparison of both similarities on the $x - y$ plot - Figure 3.7a. As many species have similar lifetimes, these are often situated within the same temporal space, which can make it hard to visually or interactively separate them. To overcome this, it is possible to convert the scatter plot into a force-simulation, Figure 3.7b. Here nodes repulse each other and are attracted to their original location. This expands the graph and prevents overlapping nodes. In doing so it is possible to interactively query the pairs of nodes which are represented by each point if required.



(a) Original (b) No-overlap

Figure 3.7: Showing the evolution from the original overlaid locations, Figure 3.7a to the slightly more accessible (interactively) Figure 3.7b

Using a Kernel density plot it is seen that both cosine and euclidean distances have a similar distribution of points for the chosen simulation. The agreement of both metrics suggests a similarity between both the lifetime values and their change over time for simulation. This is in agreement of with the $x - y$ plot of the species. In selecting species that are part of the same initial cluster and have a high agreement between both similarities it is possible to gauge the suitability for two species to be lumped together.

3.0.21 A Quick Concnentration Comparison

Having described how the similarity distances work, the best and worst pairings are shown. The results in Figure 3.8 - a log10 ensemble of the concentrations for the 300 sumulations used in the results section. Here the best matches both have a flat decay curve, whereas the the worst parings are between photolytically influenced species with a pronounced diurnal profile compared to ones with a flat loss curve.

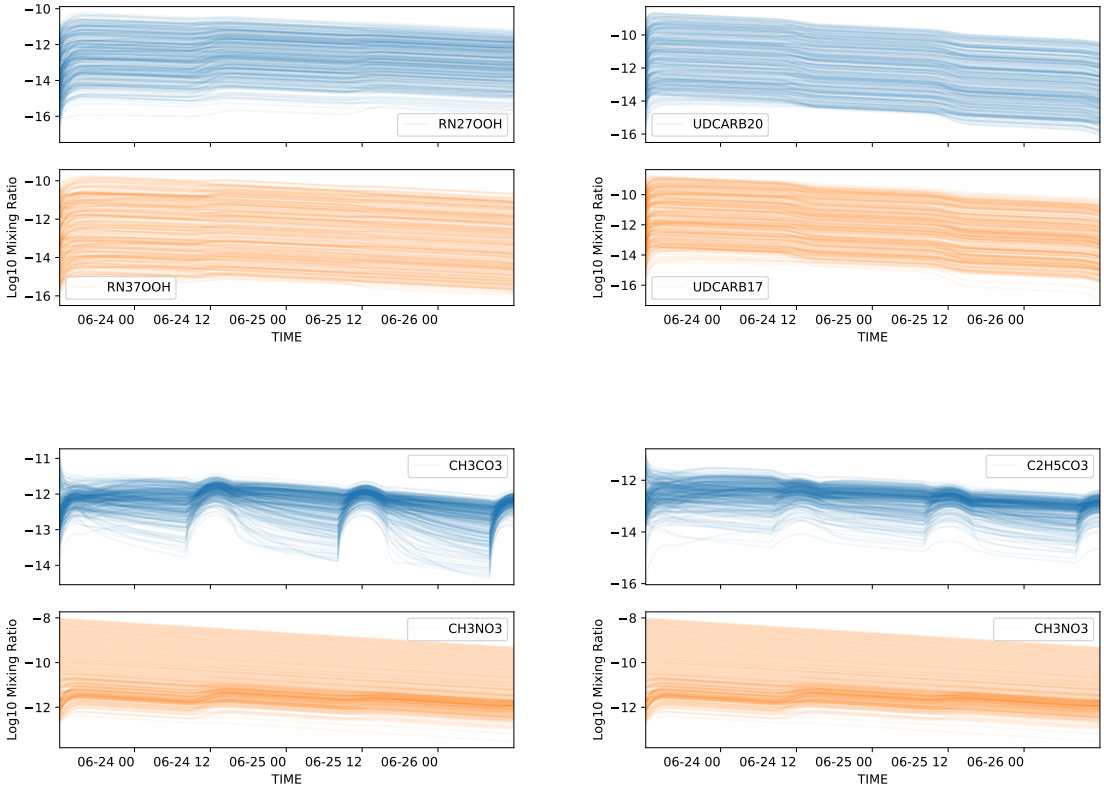


Figure 3.8: **Comparing the best (a-b) and worst (c-d) species combinations using the combigned similarity metrics.**

3.0.22 Results

In order to get a representation of the mechanism we run 300 randomly initiated scenarios. The experiemental setup is one such that it is possible to add more datapoints at a later date. From each simulation the no diagonal elements of the jacobian are used to construct a graph representative of the aggregated hourly means of the simulation output. Each of these graphs is then run through the infomap algorithm and a grouping/clustering produced. To select the best possible grouping, each infomap is run 100 times, where the result with the best fit (shortest codelength) is taken - this is an optional parameter on the algorithm.

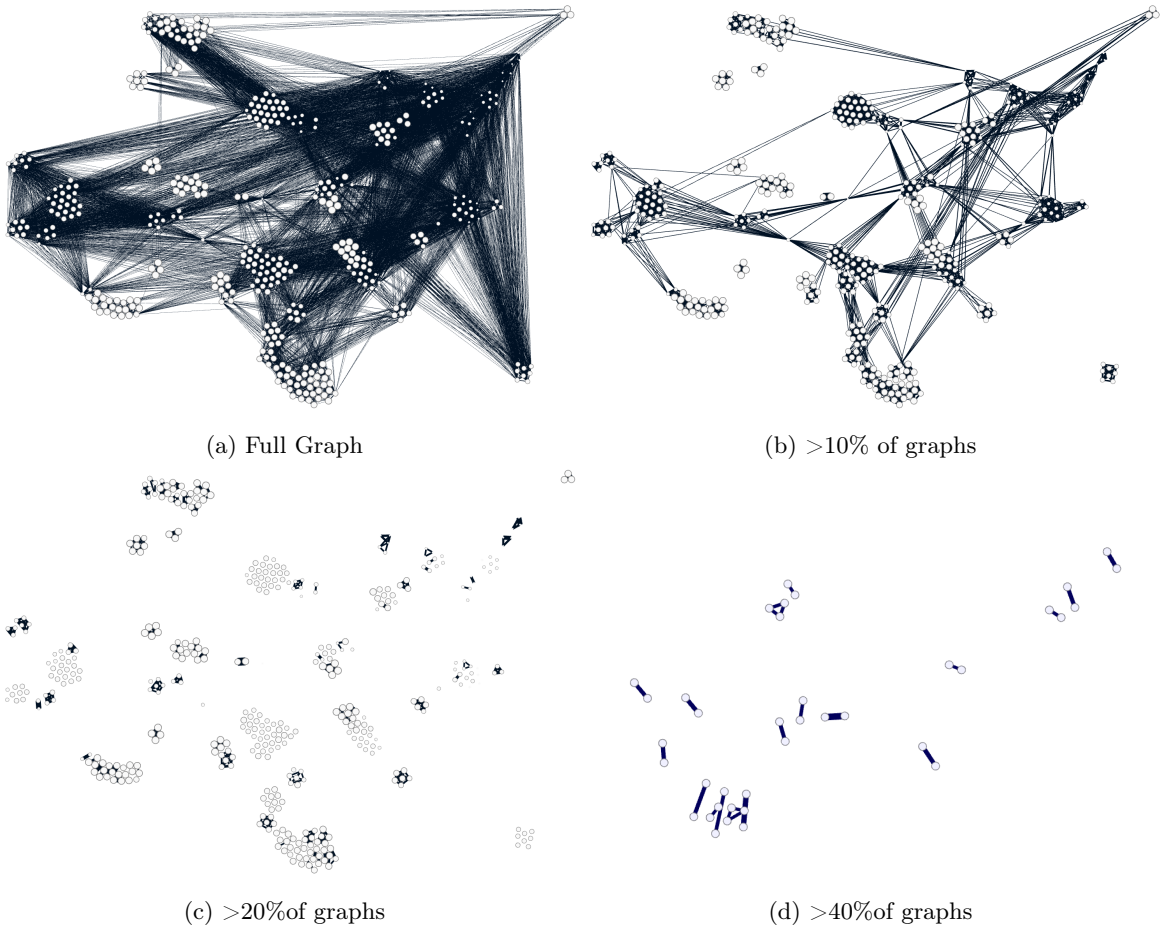
3.0.23 The Co-Grouping Network

To aggregate the groupings produced by each algorithm an $n \times n$ matrix is created for each of the n species in the mechanism. This is treated as a graph relational matrix, whereupon if species A is in the same group as species B, then a link (or value +1) is added to the [A,B] (A $\longrightarrow$ B) and [B,A] (B $\longrightarrow$ A) column. Using this matrix format it is possible to then generate a graph showing the

relationship between species that were clustered in the same group.

This relational matrix can then easily be converted into the network format: Figure 3.9a. Starting with this it is then possible to filter edges below a certain weight, Figure 3.9b-d. Finally isolates (nodes with no links) are removed, leaving only those clusters where each species has a strong relationship between every other.

In the context of this section we select only relationships that appear in over 45% of all the clustered simulation runs. The reasoning behind this is that there may exist a set of pairing that only appear either during the day or night.



(a) Full Graph

(b) >10% of graphs

(c) >20%of graphs

(d) >40%of graphs

Figure 3.9: **Filetering the infomap clustering relationship matrix/graph** How the clutering relationship network changes as weak links (links between species which do not appear in many of the infomap groupings) are removed.

3.0.24 Comparing Daytime And Nightime Groups

In determing a group of species which are commonly custered together in most simulation results, we are next interested in seeing if these groups change with day or night.

To do this we use an alluvial diagram. This is a cross between a parallel-line plot and a sanky diagram,

and is particularly suitable for showing the changes in clusters within a temporal network, [Rosvall and Bergstrom, 2010].

In taking a the common clusters formed at midnight (Figure 3.10 left) and midday (Figure 3.10 right) we are able to compare these to the overall selection (all hours - middle). Here, as is expected, any parings which persist in over 45% of all the timesteps, exist in all three categories. In addition we see a selection of species which are grouped together at both 12:00 and 0:00 hours. This suggests that they may not be grouped with some of the intermediate hours, and that if the threshold of selection is lowered below 45, they may appear in the overall result. Finally a selection of species which are only grouped togehter in daytime or night time only results.

3.0.25 Determining Cluster Suitabiltiy

Having selected clusters that appear for most graphs in the network, it is now important to assess the suitability of each node for being lumped together. Using a normalised similarity matrix, we extract the values for each of the lumbed groups, Table 3.1. Here the best values are providec by the PECOH and DIEK species, Figure 3.11a. These both have linear decaying concentrations within the same order of magnitude. This is probably due to PECOH being the only precursor to DIEK, where DIEK accounts for 0.436% of its total products. This makes them a suitable candidate for lumping. $HOCH_2CO_3H$ and $HOCH_2CO_3$ make the worst possible lumping combinations. This is because the radical $HOCH_2CO_3$ is able to react with many of the inorganics, whilst $HOCH_2CO_3H$ can only dissociate into formaldehyde or react with OH to reproduce HOCH2CO3. Although these species both have differing profiles, of several orders of magnitude difference, their cyclic nature $HOCH_2CO_3H \underset{HO_2}{\overset{OH}{\rightleftharpoons}} HOCH_2CO_3$ most likely proved to trap the 'flow' of the network, producing the cluster. Additionally there are also several clusters consisting of (N)RIxxOOH and (N)RIxxO_2 species. These are genearally species formed from iso-alkanes, and both produce acetaldehyde (CH_3CHO) as a product. Here the peroxy radical ($R-O_2$) reacts with several inorganic species, producing a varying diurnal. Regardless of this the cosine similarity is still relatively small. This may be attributed to the 'flat' periods of slow decay that is experienced at nighttime (due to the reduction of available HO_2 and NO) which follow the loss trend of the peroxide ($R-OOH$) species. Since hydrogen addition and subtraction are both fast reactions forming species with the same oxidation potential (or CRI number), it makes sense that the clustering algorithm often identifies peroxides and their peroxy radical equivalent as a group.

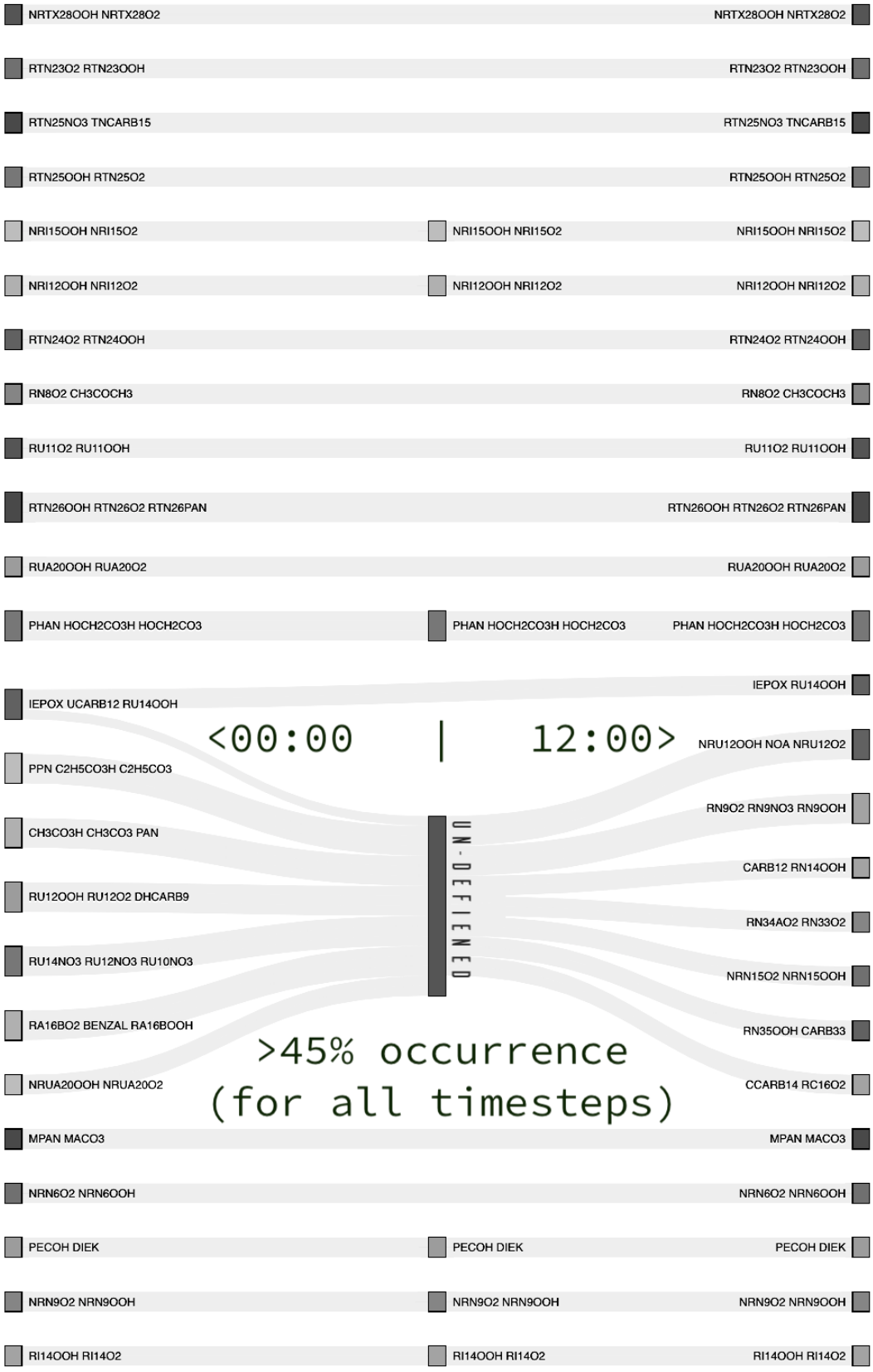


Figure 3.10: **An alluvial diagram showing the changes in clusters between noon and midnight.** On the left are all groups that apear in >45% of the midnight simulation results. On the right are groups which appear >45% of the midday results. In the middle exist the clusters extracted which appear in >45% of all runs. Here it is seen that there exist a series of species which may exist in daytime or nighttime chemistry, but do not persist between both.

Sepecies Pair	Euclidean	Cosine
NRI15OOH NRI15O2	0.4624	0.2885
NRI12OOH NRI12O2	0.4617	0.2986
PHAN HOCH2CO3	0.5103	0.9998
HOCH2CO3H HOCH2CO3	0.8350	0.9892
RI14OOH RI14O2	0.4922	0.2275
NRN9O2 NRN9OOH	0.4620	0.2818
PECOH DIEK	0.0172	0.0011

Table 3.1: A table of the **normalised** similarity values for the lumped species. Numbers closest to 1 show the worst possible paring in the mechanism, and numbers approaching 0 show the best.

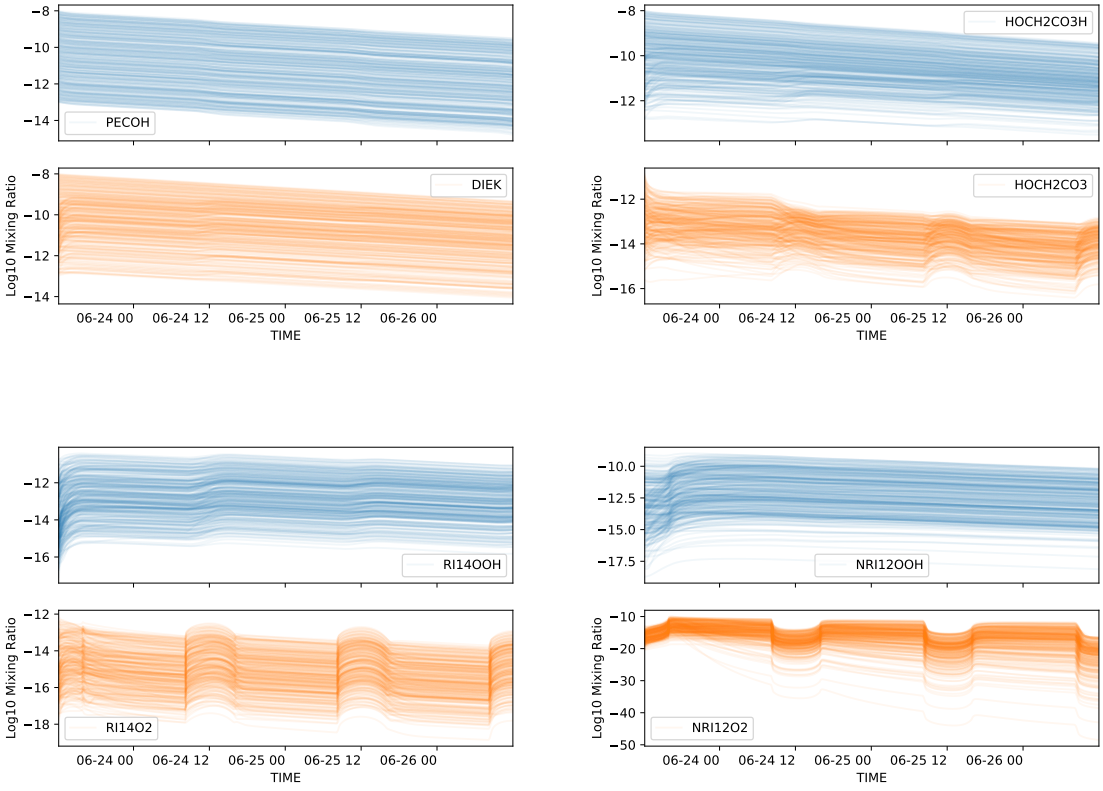


Figure 3.11: **Comparing the best (a-b) and worst (c-d) species combinations using the combigned similarity metrics.**

3.0.26 Conculsions

Graph representation of a chemical network can be used to apply graph clustering techniques. Although there are a range of methods available, the infomap method proves well suited for the use on chemical mechanism. Applying this to the CRI v2.2 mechanism we were able to sucessfully partition the chemistry into branches representing the reactions between differenct chemical structures. This was seen by exploring the hierachical structure of the InfoMap output in the form of a tree.

Additionally natural language similarity metrics can also apply to compare the temporal changes

between species lifetime. Here the Euclidean distance can compare the magnitude difference between species pairs, whilst the cosine distance looks at the angle between them. Combigned these can give us indications if the lumping of two species can prove problematic.

Finally 300 randomly initiates simulations were compared using graph clustering. Here persistent groupings were extracted and their suitability compared using the similarity metrics and their lifetimes across the entirety of the simulation.

Further work would involve the comparison of a mechanism lumped by this method to the unlumped version. The CRI v2.0 mechanism has a series of 5 further lumpings by * and can provide a useful comparison on how beneficial this method of mechanism reducticti on fares agains more traditional methods.

Bibliography

Andersson-Sköld, Y., Grennfelt, P., and Pleijel, K. (1992). Photochemical Ozone Creation Potentials: A Study Of Different Concepts. *Journal of the Air & Waste Management Association*, 42(9):1152–1158. https://doi.org/10.1080/10473289.1992.10467060.

Aslak, U., Rosvall, M., and Lehmann, S. (2018). Constrained information flows in temporal networks reveal intermittent communities. *Phys. Rev. E*, 97:062312. https://link.aps.org/doi/10.1103/PhysRevE.97.062312.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008. https://doi.org/10.1088%2F1742-5468%2F2008%2F10%2Fp10008.

Community, T. I. G.-C. (2020). Geoschem/geos-chem: Geos-chem 12.7.1. https://doi.org/10.5281/zenodo.3676008.

Cornes, P. (2008). Proposed Plans For Holme Pierrepont Whitewater Course. https://hppconcern.wordpress.com/2008/08/04/proposed-plans-for-holme-pierrepont-whitewater-course/.

Dodge, Y. (2008). *Latin Square Designs*, pages 297–297. Springer New York, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_223.

Emmerson, K. M. and Evans, M. J. (2009). Comparison of tropospheric gas-phase chemistry schemes for use within global models. *Atmospheric Chemistry and Physics*, 9(5):1831–1845. https://www.atmos-chem-phys.net/9/1831/2009/.

Everett, M. G. and Borgatti, S. P. (1994). Regular equivalence: General theory. *The Journal of Mathematical Sociology*, 19(1):29–52. https://doi.org/10.1080/0022250X.1994.9990134.

Fortunato, S. (2010). Community Detection In Graphs. *Physics reports*, 486(3):75–174. http://www.sciencedirect.com/science/article/pii/S0370157309002841.

GEOS-Chem (2020). Geos-Chem Publications. http://acmg.seas.harvard.edu/geos/geos_pub.html.

Jenkin, M. (2019). Http://Cri.York.Ac.Uk . Online.

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.

Lu, H., Halappanavar, M., and Kalyanaraman, A. (2015). Parallel Heuristics For Scalable Community Detection. *Parallel computing*, 47:19–37. http://www.sciencedirect.com/science/article/pii/S0167819115000472.

Mahajan, S. (2008). The Art Of Approximation In Science And Engineering. *MIT OpenCourseWare*. http://web.mit.edu/6.055/book/book-draft.pdf.

Mckay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61. https://amstat.tandfonline.com/doi/abs/10.1080/00401706.2000.10485979.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113. https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

Oran, E. and Boris, J. (1991). Numerical approaches to combustion modeling. progress in astronautics and aeronautics. vol. 135. *U.S. Department of Energy Office of Scientific and Technical Information*.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106. https://link.aps.org/doi/10.1103/PhysRevE.76.036106.

Rosvall, M., Axelsson, D., and Bergstrom, C. T. (2009). The map equation. *The European Physical Journal Special Topics*, 178(1):13–23. https://doi.org/10.1140/epjst/e2010-01179-1.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123. https://www.pnas.org/content/105/4/1118.

Rosvall, M. and Bergstrom, C. T. (2010). Mapping Change In Large Networks. *PloS one*, 5(1):e8694. http://dx.doi.org/10.1371/journal.pone.0008694.

Turányi, T. (1990). Sensitivity Analysis Of Complex Kinetic Systems. Tools And Applications. *Journal of mathematical chemistry*, 5(3):203–248. https://doi.org/10.1007/BF01166355.

Turanyi, T. and Tomlin, A. (2015). *Analysis Of Kinetic Reaction Mechanisms*. Springer. http://eprints.whiterose.ac.uk/84294/.

Turányi, T. and Tomlin, A. S. (2014). *Reduction Of Reaction Mechanisms*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44562-4_7.

Turiinyi, T. (1990). Reductton Large Reactton Mechantsms. *New journal of chemistry = Nouveau journal de chimie*, 14:795–gO3. http://garfield.chem.elte.hu/Turanyi/pdf/14_Turanyi_NJC_1990.PDF.

Vajda, S., Valko, P., and Turainyi, T. (1985). Principal component analysis of kinetic models. *International Journal of Chemical Kinetics*, 17(1):55–81. https://onlinelibrary.wiley.com/doi/abs/10.1002/kin.550170107.

Whitehouse, L. E., Tomlin, A. S., and Pilling, M. J. (2004a). Systematic reduction of complex tropospheric chemical mechanisms, part i: Sensitivity and time-scale analyses. *Atmospheric Chemistry and Physics*, 4(7):2025–2056. https://www.atmos-chem-phys.net/4/2025/2004/.

Whitehouse, L. E., Tomlin, A. S., and Pilling, M. J. (2004b). Systematic Reduction Of Complex Tropospheric Chemical Mechanisms, Part Ii: Lumping Using A Time-Scale Based Approach. *Atmospheric Chemistry and Physics*, 4:2057–2081. https://www.atmos-chem-phys.net/4/2057/2004/acp-4-2057-2004.pdf.

Zhou, H. (2003). Distance, dissimilarity index, and network community structure. *Phys. Rev. E*, 67:061901. https://link.aps.org/doi/10.1103/PhysRevE.67.061901.

Chapter 4

Computational Learning, Visualisation and Clustering:

Learning species structure using unsupervised machine learning.

"So, in the interests of survival, they trained themselves to be agreeing machines instead of thinking machines. All their minds had to do was to discover what other people were thinking, and then they thought that, too."

- Kurt Vonnegut, *Breakfast of Champions*

4.0.1 Introduction

Historical Significance

The established process of trial and error has always underpinned our survival [Noble, 1957]. Babies are born to rely on a set of sensory reflexes and a framework for physical reasoning [Baillargeon and Carey, 2012], and with these, we develop methods to navigate the influence of change within a physical, and auditory space [Lynch, 2011]. This method of decision making is reflected in our adult lives with ideas and actions being limited in choice by our intuition and experience [Descartes and Lafleur, 1960]. In science, we apply a methodological framework consisting of a continuous assessment of scepticism, educated guessing (hypothesising) and rigorous practical testing. Specialists accrue years of practical and theoretical knowledge within a narrow field and can identify areas of potential gain and futility. Yet even with all prior experience, the discovery of new and untested techniques involve the tortuous traipsing through a sea of uncertainty. Such a methods sometimes prove fruitful, through accidental discoveries of items such as x-rays, penicillin... [Roberts, 1989]; finding novel applications for existing methods such as optical tweezers for chemistry or the abstract field of maths utilised by Einstein [REF], but more often than not end in the constant evolution of a pre-existing project with no apparent result.

Theory And Simulation In Science

Until recently much of the experimentation possible was limited by resources, levels of knowledge available technology. With the increase of computation power, we have been able to not only increase our understanding but also run theoretical simulations to guide exploratory efforts with an impact on real-world applications [Oliveira et al., 2006; T. Leube et al., 2018; Morozov, 2016; Yu-ChenLo, 2018]. However, as our ability to record and produce data increases, the need for the scientific method diminishes [Anderson, 2008]. Here the application of ‘big data’ tools and algorithms can provide insights and correlations much more compelling than the predictive capabilities of constantly changing models - “Since all models are wrong the scientist cannot obtain a "correct" one by excessive elaboration” - Box [1976]. As our level of attainable technology increases, so does the complexity of the data collected. New datasets tend to be large, complex and highly multivariate. Although this dramatically improves the quality of science, the difficulty lies in trying to represent it in such a way that we may successfully access the reliability of the results. Since simple bar and line graphs are no longer applicable, one solution falls within a class of unsupervised machine learning techniques called dimensionality reduction (DR).

Chapter Aims

In **??**, we looked at visual representation as a way of understanding complex systems. **??** showed that the chemical properties could be inferred (visually) from the node-link graph structure of a mechanism. Similarly, Chapter 1 and **??** located the presence of important species and clusters of similar properties by applying mathematical algorithms to the graph network. As opposed to attempting to visualise complex data, this chapter looks at learning the structure of a chemical species and simplifying it into two dimensions. Here it is possible to extract key features of like-groups through the use of vector clustering, which unlike the graph clustering in **??** works by determining the density between points on a plane.

The chapter begins with the introduction of the chemical system, and the various methods for representing species structure within it (Subsection 4.0.2). Next, we define the dimensionality reduction methods, which are to be used to simplify the inputs above (**??**). This is followed by a brief overview of the visualisation methodology (**??**). Finally, all three sections are combined to produce a set of result and conclusions about the use of DR to identify species structure.

4.0.2 Species of the MCM and ways to represent them.

The master chemical system (as defined in all previous chapters), represents our foremost knowledge of gas-phase chemistry within the troposphere. **??** shows that information about a species structure is encoded within its reactions, much of which can be attributed to the well-defined construction protocols.

This section explores the different methods of representing a species structure, intending to provide a machine built algorithm with the highest amount of information about each species and its functionality. A range of input types will be evaluated against several dimensionality reduction algorithms to isolate which chemical properties are most 'picked up'.

4.0.3 Input Generation

The MCM provides species information in the form of a species 'smiles' (Subsubsection 4.0.5.2) and the IUPAC InChi string [Heller et al., 2013]. Within this chapter, we use only the smiles string, which is either manually processed using regular expressions or with the aid of pythons RDKIT package [Landrum et al., 2019]. There are seven different methods for representing the chemistry; these are outlined below.

4.0.4 Manual Categorisation

Reactions within the MCM are determined by a set of rules (PROTOCOL SECTION). These mimic the process a chemist my discover new species and often rely on the bond availability and functionalisation of a species. Since the present functional groups are the benchmark of whether a DR algorithm has successfully separated species structure, it makes sense to run a unit test using the known functional groups of a species as the input.

To generate the functional groups the regular expressions in Table 4.1 are used[1] on the smiles strings (described in Subsubsection 4.0.5.2) for each species. In extracting the functional groups, we can plot the likeliness a species with a certain group is likely to have another using a chord diagram - Figure 4.1. Since most species contain a multitude of functional groups, the separation of these into 'tidy' clustered groups seems unlikely.

PAN	`C\\(=O\\)OON\\(=O\\)=O$|^\\[O-{0,1}\\]\\[N\\+{0,1}\\]\\(=O\\)OOC|O=N\\(=O\\)OOC\\(=O\\)|C\\(=O\\)OO\\[N\\+{0,1}\\]\\(=O\\)\\[O-{0,1}\\]`
Carb. Acid	`[^O](C\\(=O\\)O$|^OC\\(=O\\))`
Ester	`[\^O](C\(=O\)O\b|OC\(=O\))C`
Ether	`(\([\^O=]+\))*C(\([\^O=]+\))*O(\([\^O=]+\))*C(\([\^O=]+\))*`
Per. Acid	`c\\(=O\\)OO$|^OO\\(=O\\)C`
Nitrate	`O(NO2\b|NOO\b|N\(=O\)=O|\[N\+\](?:\[O-\]|\(=O\)){2})`
Aldehyde	`C=O$|^O=C`
Ketone	`C\(=O\)C`
Alcohol	`CO\\b|(?=^\\b)(?!^\\[)CO.|(?=^\\b)(?!^\\[)OC.|\\(O\\)|C\\)O(\\b|[^O]`
Criegee	`\[O-\]\[O\+\]`
Alkoxy rad	`\[[\/]{0,1}CH{0,1}\]|\b[\^O]\[O\.{0,1}\]`
Peroxyacyl rad	`\\ w\(=O\)O\[O\.{0,1}\]`

Table 4.1: CHECKKKKKKK!!!!!!!!! A set of regular expressions that may be used to determine the number of occurrences of a functional group within a SMILES string.

[1] To see the structure of each functional group type, go to **??**.

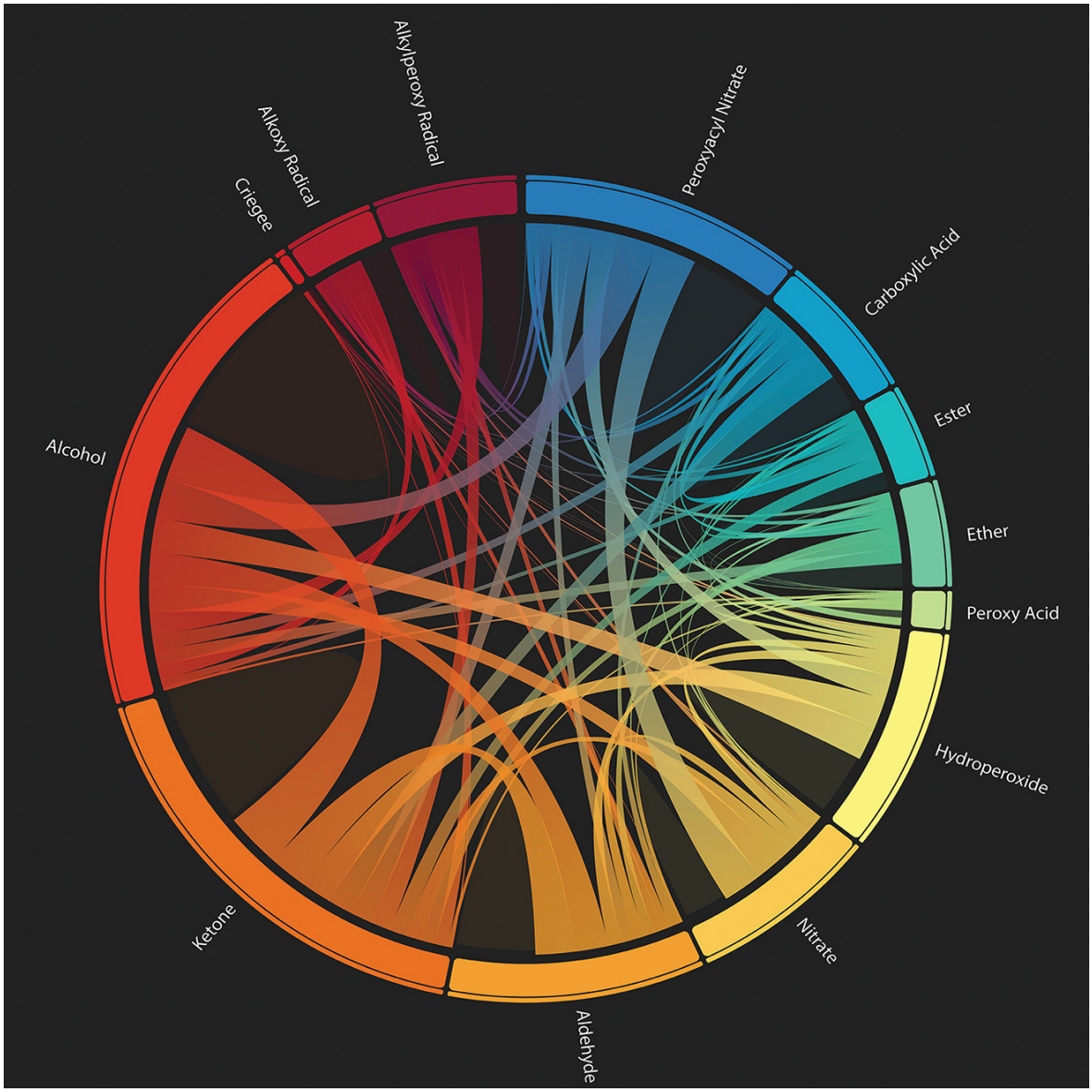


Figure 4.1: **The multifunctionality of the MCM.** A chord diagram showing the functionalisation of a species within the MCM. Arc sizes represent what percentage of all functional groups in the MCM mechanism a group contains. Translucent areas of no outwards links represent species with multiples of a certain functional group, of which Alcohols and Ketones have the most. Source: [Ellis, 2019]

4.0.5 Tokenization

As computer algorithms are unable to understand words or their meaning, we have to first categorise the data into groups. Tokenisation is the conversion of a string into characters and representing them with a numerical equivalent. In doing so, a string of characters can be converted into a numerical vector, allowing for its representation in a latent vector space. Within our input selection, we have two sets of inputs we can convert. These are the species names, and their smiles string representation.

4.0.5.1 Species Names

In **??** it was shown that the dedicated species names for species in the CRI mechanism were often representative of their structural properties. This adage also applies for the MCM, where an intuitive naming convention is used. This is often derived as part of the construction protocol, where a species names reflect its own, or its precursor's structure (which it will have at least in-part inherited).

Although this is not the most robust method of defining the structure, it allows for a straightforward test of the algorithms, for which the user can quickly compare the human-readable output.

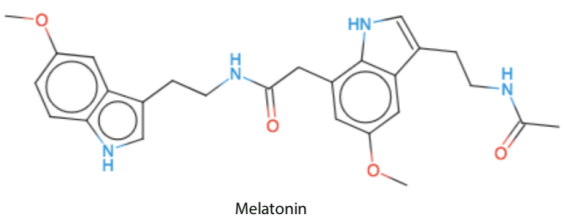
4.0.5.2 SMILES Strings

Smiles ('Simplified Molecular-Input Line-Entry System') provide a human-readable representation of the molecular structure, [Weininger, 1988]. They offer a linear human-readable description of the chemical composition within a molecule - making it easy to visually check the construction of a species without any additional work. Besides, their role in generating the molecular fingerprints in Subsection 4.0.7 makes it a useful comparison to make when evaluating methods of structure representation.

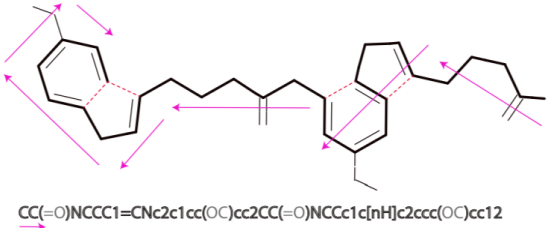
Construction Methodology of SMILES strings

The construction of a SMILES string happens in three parts:

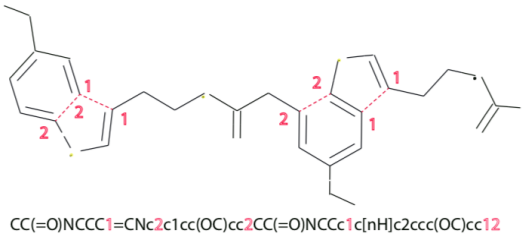
1. The smiles string is built by creating the longest possible chain to form a molecule backbone. Figure 4.2b
2. This may within itself contain aromatic rings denoted by the lowercase carbons and a number corresponding to the location of each break cycle. Figure 4.2c
3. Finally all the functional groups and branches attached to the main backbone are added. These are nested within the parenthesis to show that they are not part of the skeletal backbone. Figure 4.2d



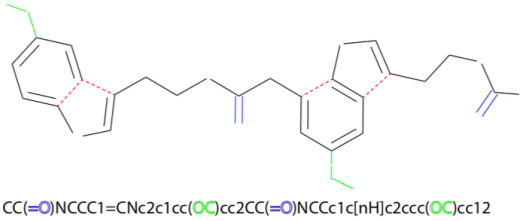
(a) Structure of Melatonin



(b) Step 1 : Building the C chain backbone.



(c) Step 2 : Aromatic Rings



(d) Step 3 : Functional Groups

Figure 4.2: **Construction process of a smiles string.** The example compound is Melatonin. Although this does not exist within the atmosphere, it provides a clear example of the smiles string methodology. Figure 4.2a is made using smiles drawer: [Probst and Reymond, 2018]

4.0.6 Graph Inspired

?? - **??** have shown the role of graphs in revealing network properties and structure. Graphs in themselves can simplify relational data into two/three dimensions for visualisation and algorithmic clustering. Continuing this trend, we can represent a species structure in the form of a graph (Subsubsection 4.0.6.1), as well as converting the structure of a mechanism for dimensionality reduction (Subsubsection 4.0.6.2)

4.0.6.1 The Species Graph (Fingerprint)

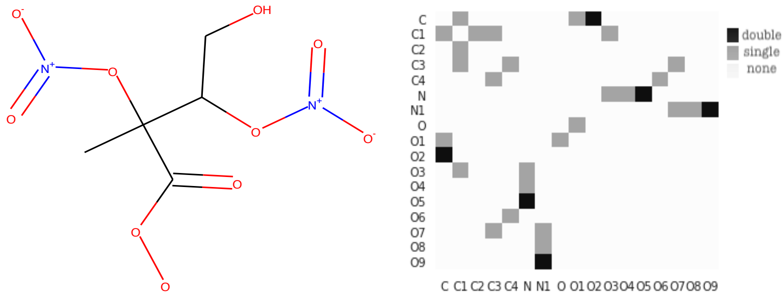
The structure of a species has long represented using a graph-like layout, **??**. It, therefore, follows that other methods for representing the graph structure would also apply. One such way is the use of an adjacency (or relational) matrix to describe the relationships between atoms and bonds in a species. Such a methodology is already used in the construction of bond and z-matrixes [Aumont et al., 2005; Parsons et al., 2005].

The construction of a structure matrix/graph begins with a chemical species. Here the relationships between atoms (Figure 4.3b) is converted into an adjacency matrix (Figure 4.3c). However, since species have different numbers of each atom, a template allowing us to compare different graphs is required. To do this a maximum occurrence table (Figure 4.3a) is created. Here, for example, BCARY $C_{15}H_{24}$, a sesquiterpene contains the most carbon atoms of any species within the MCM. This universal matrix is now able to contain any possible combination of atoms in a species.

As machine learning algorithms only vectors as an input, it is possible to decompose the 37^2 element adjacency matrix into rows, which can then be joined together, Using this method we create a one-dimensional array (vector) of 259 elements (518 bytes) to represent our species.

Atom	Max
C	15
Cl	4
O	12
N	3
S	1
BR	2

(a)



(b) (c)

Figure 4.3: **Constructing a graph from species structure.** (a) shows the maximum number of times an atom occurs for any single species in the MCM. (b) depicts the graph-like chemical structure of INB_1NBCO_3. This is a highly processed species stemming from Isoprene, and this makes for a good example of the bond matrix. Finally, a matrix representing the bonds in INB_1NBCO_3 is created from the maximum possible occurrence matrix from (a). For simplicity, empty row/column pairs have been removed to produce (c). This matrix will always be symmetrical as the bonds do not have a direction.

4.0.6.2 Node Embeddings (Node2vec)

?? and Chapter 1 showed that the underlying structure of a chemistry mechanism graph contains information about the species and reactions within it. In Figure 4.4 colour represents the ratio of potential oxidation of a species. Here as emitted species become progressively more processed, the number of bonds which may be oxidised diminishes (lighter colours near the centre) until they eventually form carbon dioxide and water.

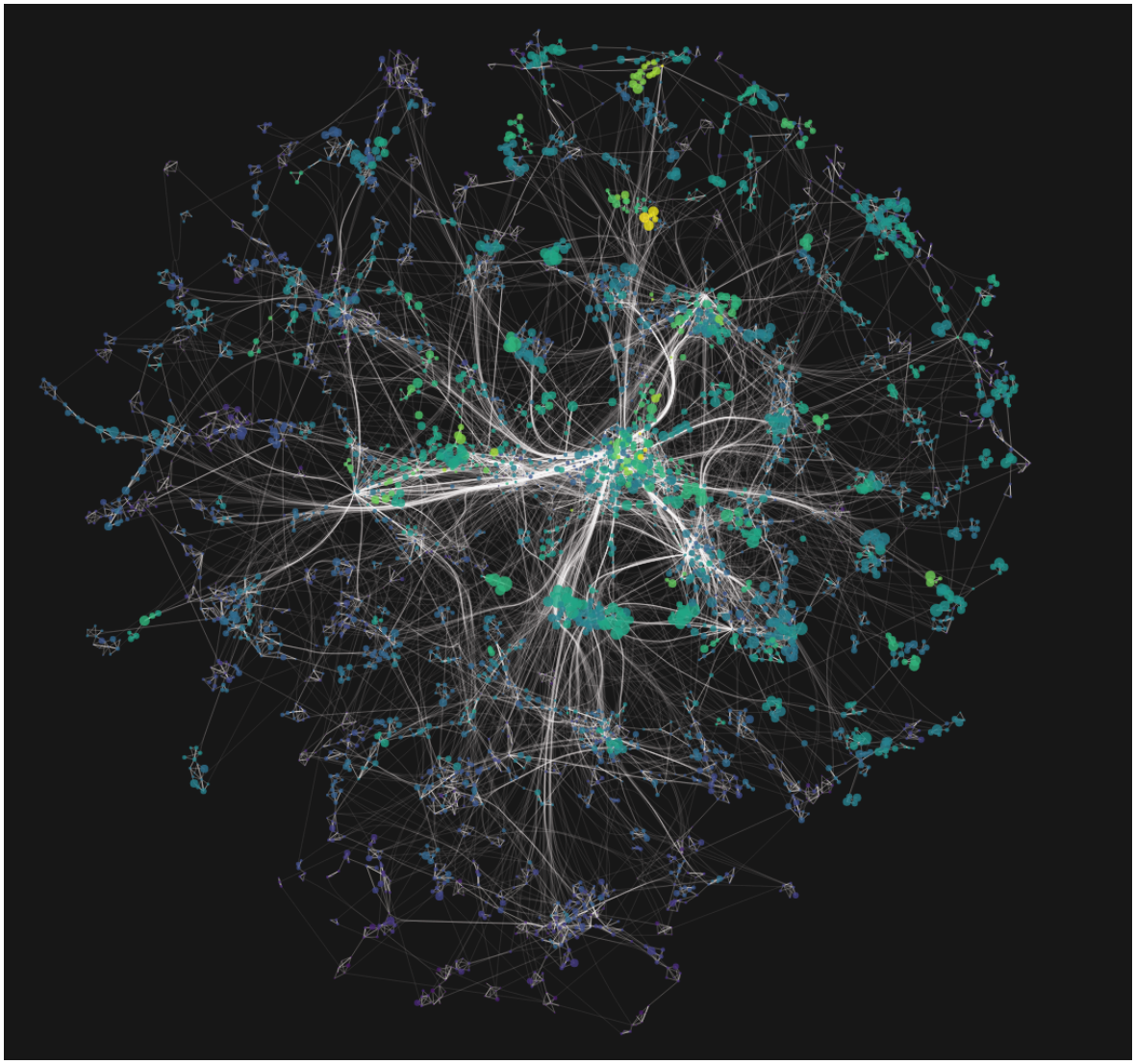


Figure 4.4: **The graph of an MCM subset representing the chemistry within Beijing.** Here colours show the increase of O−C ratio as species are oxidised (lighter). All emitted species ultimately tend towards carbon monoxide which is at the centre of the graph.

This type of structural information can be extracted through the use of a natural language processing package capable of transforming a graph into a vector - node2vec [Grover and Leskovec, 2019]. Since this may also be used for dimensionality reduction, it is described within the next section (Subsection 4.0.14).

4.0.7 Molecular Fingerprints

In the field of chemical informatics, molecular fingerprints (or structural keys) are used to encode and query structural properties of species. Their binary representation makes them suitable for dimensionality reduction and the exploration of fo chemical space (a type of property space constructed using pre-determined features and boundary conditions).

Here species properties are often split into structural and psyico-chemical groups - which has used such as the discovery of natural analogues (which circumvent problems such as intolerances in medicine [Spahn et al., 2017]). Although there exist many different types of molecular fingerprints, the two main ones that will be explored are molecular quantum numbers (MQN) and the molecular access system (MACCS).

4.0.7.1 Molecular Quantum Numbers (MQN)

In chemistry the shape, phase and electron occupancy of an atom may be described through the use of four quantum numbers: the n principle quantum number, I angular momentum quantum number, M_i magnetic quantum number and M_s spin quantum number. The rationalisation of elements based on their structure, and by consequence reactivity, has led to the most iconic tool of the modern-day chemist - the periodic table, where increasing atomic numbers follow the principal quantum number [Wang and Schwarz, 2009]. In representing a molecule as a set of 42 quantum numbers, MQN fingerprints produce a multi-dimensional mapping of atom, bond, polarity and topology count [Nguyen et al., 2009].

4.0.7.2 Molecular ACCess System (MACCS)

MACCS keys are a 164[2] bit structural keys formulated through answering a series of structure-related questions. Developed by MDL Information Systems [, MDL], their main purpose lies in being a SMILES Arbitrary Target Specification (SMARTS) system for substructure searching. However, their distinct structure key format makes them highly suitable for similarity detection. In many cases, the optimised version of MACCS keys is cited ([Durant et al., 2002]), although most use cases exploit a variation of the undocumented 166bit keys. We use the implementation presented by [Landrum et al., 2019; rdkit, 2019] for all molecular fingerprints in this section.

4.0.8 Dimensionality Reduction Methods

In the last section, we described several methods in which the chemical structure of a species could be encoded for direct comparison. However, since each input consists of a multitude of elements, it is still not a simple task to determine the differences and similarity between all species in mechanisms. Dimensionality reduction is the process of reducing the number of random variables and only presented a set of principal values, by mapping a high-dimensional space into a low-dimensional one [Roweis

[2]They are 166-bit keys, although there is no real agreement to what the 44th keys' purpose is, and therefore it is often omitted. Within RDKIT this is denoted by a ? [rdkit, 2019].

and Saul, 2000]. This allows us to flatten a multivariate input into the two dimensions required for a simple scatter plot.

In this section, we begin by explaining the data preparation required for dimensionality reduction (**??**) before describing the different possible methods of reducing the dimensions of a dataset.

4.0.9 Preperation Of The Data

Real-world data is rarely preformatted in such a way that it can be used directly within a computational model. Often values need to be cleaned and corrected to be fit for purpose. In the interest of completeness, the two main methods of data adjustment for machine learning are outlined below. These are normalisation and standardisation.

Normalisation

In the data is without (dimensionless) or of a single unit, it is possible to rescale the data between a range - most commonly 0,1. In doing so it is possible to interpret the importance of value in contrast to the largest recorded value. This gives us a percentage scale spanning the range of the data. Such a range is useful in the definition of colourmaps and describing the importance of value relative to the dataset. To rescale a dataset we shift the minimum value to zero, then divide by the new maximum of the dataset (Note this is equivalent to the range of the unshifted dataset.)

$$n(x_i) = \frac{x_i - \min_x}{\max_x - \min_x} \tag{4.1}$$

Standardisation

If the components we wish to compare are of different units or are expressed with a different scale, normalising them would not produce meaningful data. Instead, it is possible to standardise the data by looking at each points deviation from the mean. Here the variation of the mean for a dataset is divided by the standard deviation to produce a value between {-1,1}, Equation 4.2. In statistics this is known as the 'z-score'[3]

$$z(x_i) = \frac{x_i - \mu_x}{S} \tag{4.2}$$

[3]Possibly because of the American spelling of standardi**Z**ation?

4.0.10 Principle Component Analysis

One of the most well-known dimensionality reduction methods is the determination of the principal components through the use of Principal Component Analysis (PCA). PCA increases the readability of a dataset by creating a set of new uncorrelated variables which maximise the variance [Jolliffe and Cadima, 2016].

PCA works on the assumption that components within a dataset are linear combinations of each other. By simplifying these linear combinations, it is possible to identify the elements which explain the most variability in a dataset - these are the principal components.

A more straightforward interpretation of this would be to adjust the direction of each axis of the data, such that its projection has the most prominent variability. In doing so, it is possible to determine which components contribute the most to changes in the dataset [F.R.S., 1901; Hotelling, 1933]. An example of this is seen in Figure 4.5, where the second component of the original data can be removed with little effect on the overall result of the data. Such methods have applications in compression and signal filtering [Hernandez and Mendez, 2018; Hamadache and Lee, 2017].

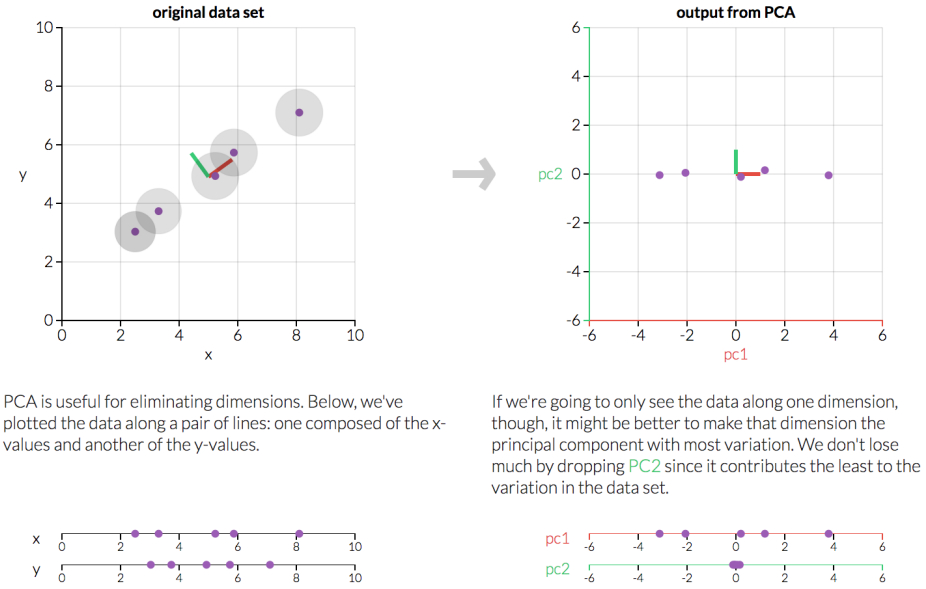


Figure 4.5: **Determining the Principal Compnent of a sample dataset.** It can be seen that in a change in axis to follow the first principal component (right), it is possible to explain most of the variation in the samle dataset (left). Source: [Powell, 2020]

4.0.10.1 Mathematical Explanation Of PCA

Note: *The basic statistics/mathematics required to understand this section is shown in* **??**. *Please read this if you are not familiar with any of the terms below.*

The mathematics behind PCA consists of first calculating the covariance matrix - an $n \times n$ matrix

outlining how strongly each variable changes with every other. Using this we can calculate both the eigenvalues and eigenvectors of the matrix [4]. This can be done using a computational package such as numpy or scipy [Oliphant, 2006; Jones et al., 01].

We can now sort the eigenvector columns by influence using their eigenvalues—this way a feature dataset can be produced by removing vectors of low importance. The final feature dataset can now be transposed and multiplied by the transpose of the original dataset. This results in an output dataset containing each principal component of the desired dimension.

4.0.11 T-Distributed Stochastic Neighbor Embedding (T-SNE)

t-SNE is an algorithm designed with visualisation in mind [Maaten and Hinton, 2008]. Rather than representing the data through a series of linear transformations, t-SNE uses local relationships to create a low-dimensional mapping, much in the same way as a fully connected force graph, Figure 4.6. This allows the ability to capture non-linear structures in the data which cannot be accomplished through linear mapping methods (e.g. PCA).

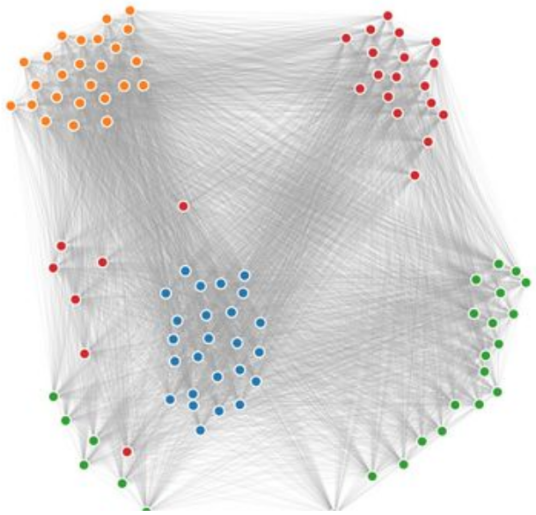


Figure 4.6: **Representing the t-SNE algorithm as a fully connected force graph.** Here each node is attached to every other node. Nodes with a strong relationship are pulled closer together than those with a weaker one.

The algorithm itself can be simplified into two parts,

1. Create a probability distribution which dictates relationships between neighbouring points
2. Recreate a lower-dimensional space following the probability distribution established in 1.

[4] These need to be unit vectors, although most packages already do this out of the box.

and is described in Subsubsection 4.0.11.1.The main reason t-SNE produces good results is that it can handle the **'crowding problem'** very well. The crowding problem is a product of the 'curse of dimensionality'. In a high dimensional space, the surface of a sphere will grow much quicker than one in a lower dimension space. This means that the higher dimension spaces will have more points at a medium distance from a certain point, Figure 4.7. When we map our data into a lower dimension, data will try to gather at its medium distance, resulting in a more 'squashed', and thus crowded, output.

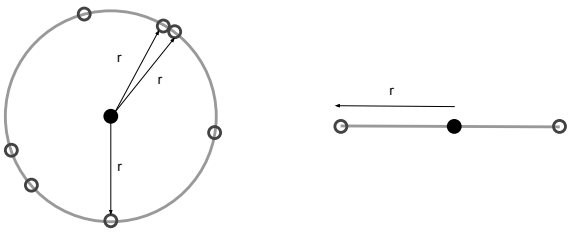


Figure 4.7: **An example of how the curse of dimensionality affects the mapping of points a certian distance from eachother.**

4.0.11.1 Mathematical Explanation Of T-SNE

In the original paper [Maaten and Hinton, 2008], the algorithm is described using the etymologic dissection of its name.

Step 1

First we begin with Stochastic Neigbour Embedding (SNE) - the distribution across neignbouring datapoints in our high dimension space. This is done by converting the high dimensional Euclidian distances between points into conditional probabilities representing their similarity:

$$p_{ij} = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\left\|x_k - x_l\right\|^2 / 2\sigma_i^2)} \tag{4.3}$$

Here $p_{i|j}$ is the conditional probability that x_i may pick x_j as a neigbour. This is proportional to the probability density of a Gaussian σ_i centered at x_i.

Perplexity Since we want the number of neighbours of each point to be similar in number and prevent a single point from having a disproportionate influence on the entire system we introduce a hyperparameter named *perplexity*. Perplexity works by ensuring that σ_i is small for points in densely populated areas and large for spare ones and can be thought of as a scale of the number of neighbours considered for any one point in the system. Generally, values between 5 and 50 are considered to give

good results, with larger perplexities taking global features into account, and by consequence smaller ones, local features.

Step 2

Now a probability distribution describing the relationship between points has been formulated, we wish to express this as a low dimensional mapping of our inputs X in terms of our output dimensions Y. Naturally, we would want to make the low dimensional mapping represent a similar (Gaussian) distribution as in Step 1. However, it often causes issues presented by the 'overcrowding problem', Subsection 4.0.11, as the gaussian has a 'short tail', and thus nearby points are likely to be pushed together. A solution to this is the student t-distribution which has a longer tail [5]:

$$q_{i|j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{4.4}$$

Note: *The definition and explination of the Student t-distribution is given in* **??**.

The optimisation of this equation is achieved through the use of *gradient decent*[6] on the Kullback-Leibler divergence **??** between distributions p and q. Here the gradient is used to apply an attractive and repulsive force on the items[7].

4.0.12 PCA Vs T-SNE, A Quick Comparison.

PCA has been around for much longer than t-SNE, and its uses are well established within the scientific community - an example of this would be the use of sensitivity analysis within mechanism reduction [Turanyi and Tomlin, 2015]. It is fast, simple and easy to use and very intuitive. The PCA algorithm works by creating a lower-dimensional embedding which best preserves the overall variance of the dataset. Clusters created from the algorithm are grouped in ways, such that they retain the highest variance of the data.

The main drawback of PCA is that it is a linear projection. If our data happened to be in a 'swiss roll' (spiral) pattern, we would not be able to 'unroll' it. The reason for this is that the PCA algorithm works by viewing the data from different perspectives, much like casting a shadow from various directions. With such an example, there is no one way we can do this that unfurls the spiral.

[5] The distribution employed is a t-distribution with only one degree of freedom and is identical to the Cauchy distribution

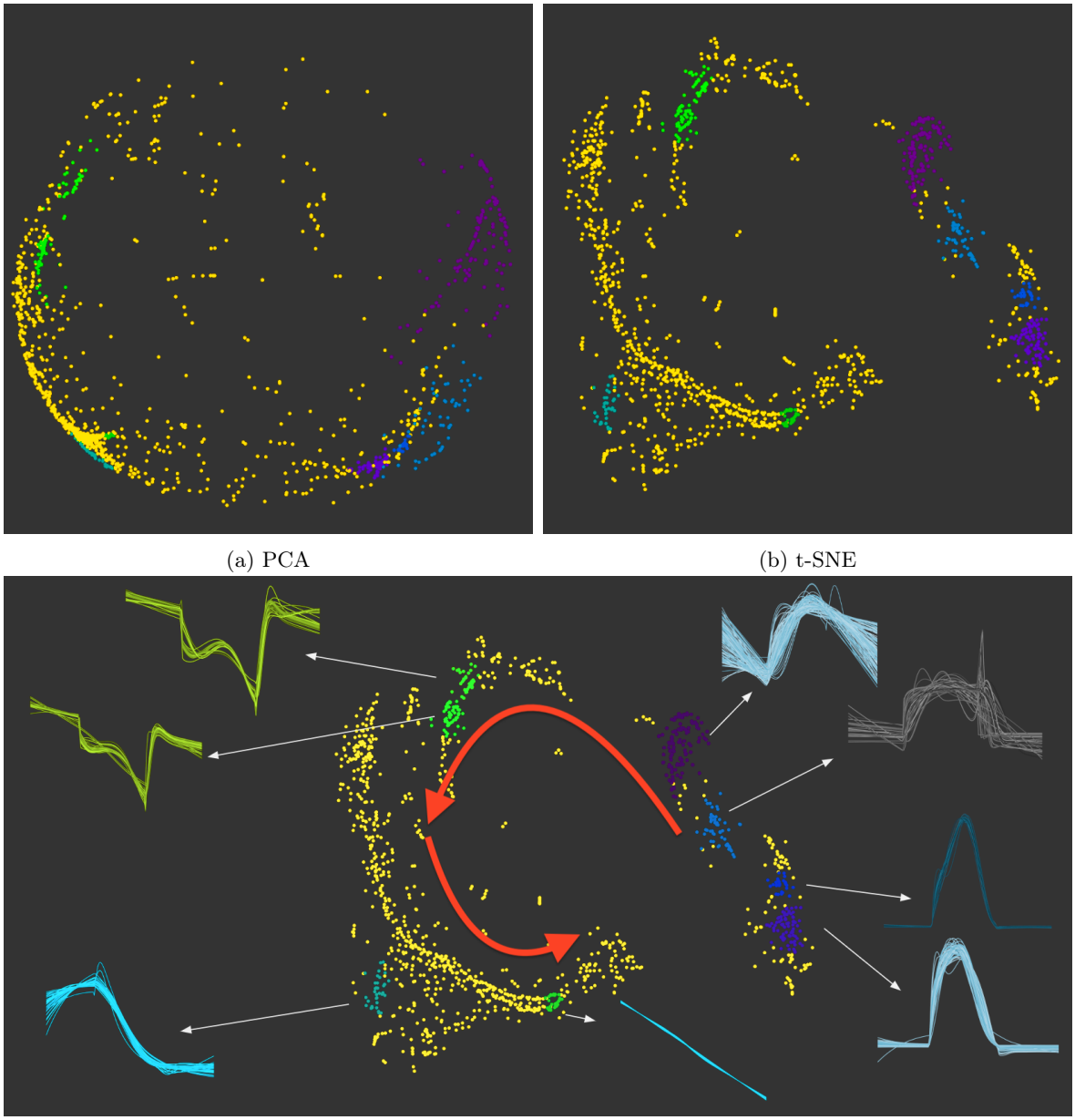
[6] **Gradient Decent** - an optimisation algorithm used to minimise a function by iteratively moving in the direction of the steepest descent. Gradient descent is used to find local minima and is defined by the negative of the gradient of the system. Its primary usage in machine learning is the updating of parameters (coefficients in linear regression and weight in neural networks).

[7] A positive gradient signifies attraction, while a negative one corresponds to repulsion.

t-SNE, on the other hand, is a relatively new method [Maaten and Hinton, 2008]. Its greatest asset is that linear projections do not limit it. Although more computationally intensive for large datasets, t-SNE produces visibly cleaner results. Unlike in PCA, t-SNE cannot be trained on additional data at a later point; however, the output clusters are more visually distinct (they have less of an overlap). Much like in a force graph, the output from t-SNE is scale-invariant. This means that while the location of clusters in a PCA reduced representation has an attributable quality, those produced by t-SNE will not necessarily contain the same information.

A box model run representative of the chemistry within Beijing was used to compare the differences between PCA and t-SNE. The aim is to classify the diurnal profiles of each species concentration (much like the cosine similarity in Subsubsection 3.0.19.2). Diurnal profiles were extracted on the third day of a spun up model initialised with initial conditions representative of the chemistry within the Beijing environment (Table 2.4). These were then standardised and converted into temporal vectors for use in the algorithms.

Figure 4.8 shows the output of both dimensionality reduction algorithms on the dataset. Different colours represent the location of clusters of similar diurnal profiles. A higher dispersion between clusters and species overlap is seen within the PCA output, Figure 4.8a. This makes it harder to distinguish species from each other or other groups around them. Since the distance between clusters within t-SNE does not hold the same mathematical meaning as PCA, the algorithm can provide a better distribution of points, creating better-defined clusters, Figure 4.8b. The concentration profile shapes for each coloured group is shown in Figure 4.8c.



(a) PCA

(b) t-SNE

(c) t-SNE with cluster outlines.

Figure 4.8: **Showing the difference between PCA and t-SNE clustering.** These figures show the clustering of a set of standardized concentration profiles (c) across two styles of dimensionality reduction: PCA (a) and t-SNE (b).

4.0.13 The Auto-Encoder (AE)

Auto-encoders are a subclass of neural networks with primary use in compressing data (dimensionality reduction). Rather than predicting a numerical output, AutoEncoders focus on the construction and deconstruction of data through the use of an encoder and decoder pair. The encoder takes an n-dimensional input and applies a compression, reducing it to the number of dimensions in the bottleneck layer. The reduced dataset is then reconstructed within the decoder. Such a process not only allows for an easy understanding of the error of the reduced data but can also be used in the filtration of

noisy or pixelated data [Leite et al., 2018; Dataman, 2019] and as an input to more complex machine learning models.

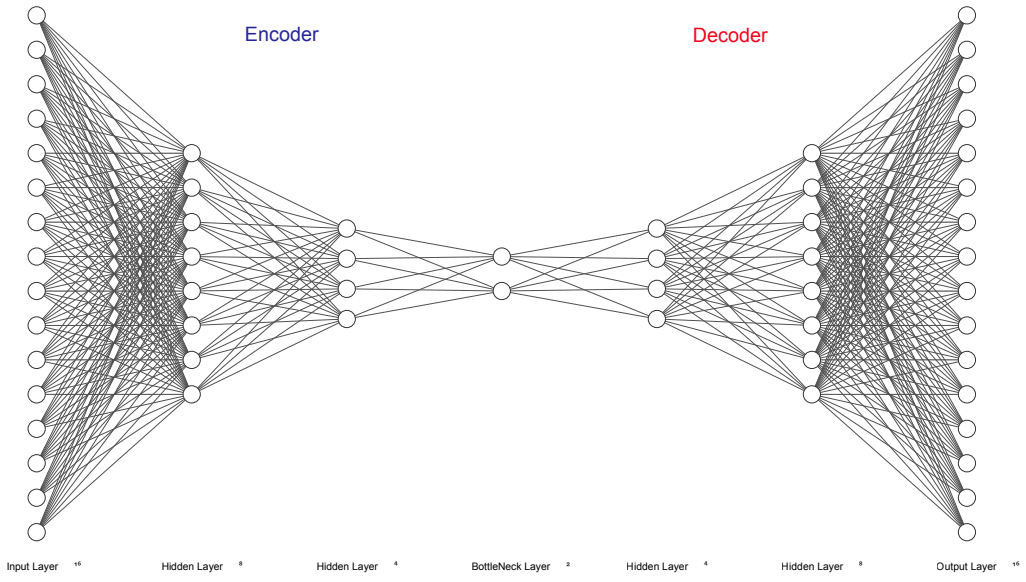


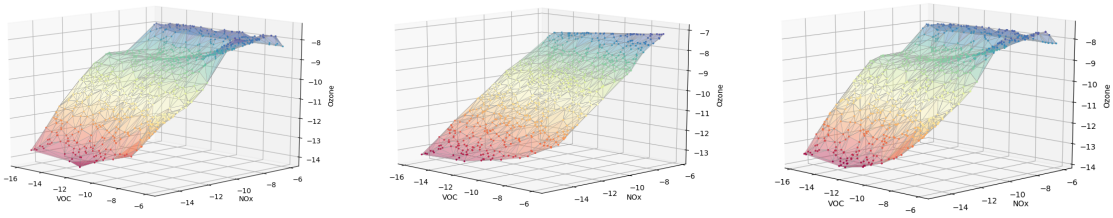
Figure 4.9: An example autoencoder structure which reduces a 16 dimentional input to 2. Draw with the aid of [Krizhevsky et al., 2012]

There are two features of an autoencoder that make it powerful. The first is the ability to sample your latent space using the decoder. The implications of this are that we can establish features that correspond to gaps between our data points - which can have its application if the data used is sparse or incomplete. Next comes the inherent non-linearity of the model. As an autoencoder is just a neural network, the amount of information passed through each link between layers is governed by an activation function. Should this activation function be linear, the reduced dimension will be much akin to a PCA decomposition. Where PCA reduces the dimensions of a dataset by discarding those with a little effect on the variance, an autoencoder opts to combine it- here the entirety of the dataset remains encoded within the links of the AE network. To decide how data flows along the edges of the network, a series of threshold (activation) functions are used for each layer. These are described in **??**.

4.0.13.1 Demonstration Of Non-Linear Activation Functions

To demonstrate the effect of these we take a sample isopleth of Methane and Ozone, reduce it to two dimensions. This is then reconstructed back into three dimensions using the DR algorithms. Figure 4.10 shows the difference between the original dataset (Figure **??**) and that of the PCA (Figure **??**) and AutoEncoder (Figure **??**) reconstructions. Here we see a loss in the non-linearity of the

original data for the PCA reconstruction. However, the use of a non-linear (tanh) activation function within AutoEncoder produces a result much closer to the original. Use of a linear activation function, however, produces a similar result to the PCA algorithm.



(a) Original (b) PCA (c) AE (Tanh)

Figure 4.10: **Comparing the result of the 2D encoding and decoding of an Ozone-NOx-Methane isopleth.** The original data (a) is reduced to two dimensions and then reconstructed back into 3D. This is done with Principal Component Analysis (b) and an AutoEncoder (c). The original isopleth is created using 300 simulations of different intial conditions: NOx (variable), Methane (variable) and Ozone (constant). These were designed using a latin hypercube and converted into a surface plot using Delaunay triangulation.

4.0.14 Node2Vec

Finally, Node2Vec is an embedding algorithm designed to generate vector representations of the nodes in a *undirected* and *unweighted* network. Although it can be used to reduce a complex network into a 2D vector (dimensionality reduction), for this experiment we shall only use it to generate a fingerprint for a species' position within a mechanism network graph - and then apply this as an input to the DR methods above. This method of input creation has been found more computationally efficient, by circumventing the need for expensive composition, in producing better predictions on network-related tasks compared to more classical methods such as PCA [Grover and Leskovec, 2019].

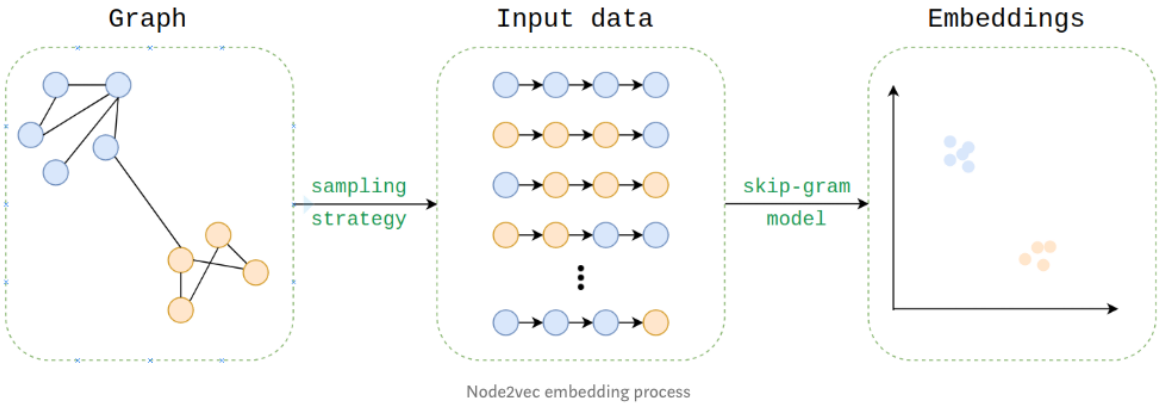


Figure 4.11: The process of converting a graph into a vector using Node2Vec. Source:[Cohen, 2018]

The process of converting the graph structure (Figure 4.11) into a numerical vector node embedding

starts by taking a series of 2^{nd} order random walks. These describe the neighbourhood of a node in the form of a set of random walk paths, much in the same way words are dependant on their neighbours within a sentence: Equation 4.5.

$$ISOPRENE \rightarrow OH \rightarrow TISOPA \rightarrow ISOPBO_2 \rightarrow TISOPA \rightarrow ... \tag{4.5}$$

This methodology allowed for the use of word2vec algorithm, converting the walk into a vector (Subsubsection 4.0.14.2)

4.0.14.1 Sentence Construction By Sampling Of A Network

The probability and path depend both on a set of arguments and a random seed provided to the model. The return and input parameters (p & q) determine how fast we explore the network and our probability to leave the neighbourhood, Figure 4.12. In a system, where the previous path is from t to v, we may calculate the probability of returning to t as $1/p$, going to a mutual node connected between t and v as 1, and viewing a new node as $1/q$. If $q > 1$ we have a high probability to end up at nodes close to t, and with $q < 1$ we are likely to explore other nodes. Additionally if we chose $p > \max q, 1$ we are less likely to return to an already visited node ($p < \min q, 1$ is likely to generate a backwards step). Since we wish to generate a 'local' view, but do not wish to return to t we select $q \geq 1$ and $p > q$ our parameters as $p = 2.0, q = 1.1$. In the case of a weighted graph (something that we are *not* exploring within this chapter) the resultant *alpha* value calculated is further multiplied by the edge weight.

To run the simulation, we use the python2 code provided by the original paper [Grover and Leskovec, 2019] with a set of 50000 random walks, each of length 9. The reasoning behind this is that we have a large graph, with a power-law like structure (where species are often heavily connected, Chapter 1).

NOTE: This process takes over a week to compute (in serial), and then the binary file containing all walks in character form approaches 10 GB, for the complete MCM.

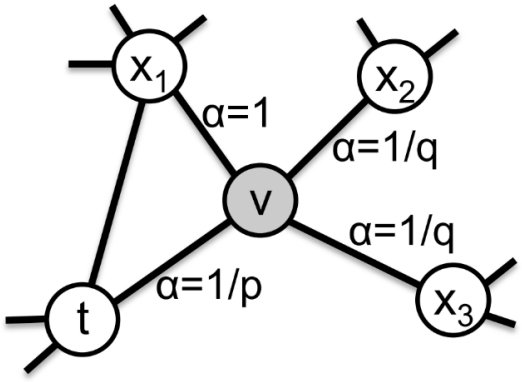


Figure 4.12: Calculation of the random walk path. Source:[Grover and Leskovec, 2019]

4.0.14.2 Word2Vec

Once we have constructed our random path 'sentences' (e.g. Equation 4.5), we can make use of Googles word2vec algorithm [Mikolov et al., 2013]. This is similar to an auto-encoder in many regards; however, the algorithm looks at neighbouring words (or species) in the corpus rather than learning word embeddings using reconstruction. This form of representation has found many uses beyond the realm of natural language processing. Some of these are objects, people, code, tiles,genes and graphs [Lynch, 2011; People2Vec, 2019; Alon et al., 2019; Jean et al., 2018; Du et al., 2019; **?**].

4.0.15 Summary

There exist several methods of reducing a complex dataset into a smaller one. PCA is the simplest method to understand but is constrained to linear decompositions. AutoEncoders can have both a linear and non-linear response, based on the activation functions that they use, and t-SNE applies a non-linear grouping which mimics a complete force-directed graph.

Having defined each method, we next explain how they will be evaluated (Subsection 4.0.16), before applying them to the MCM in Equation 4.5.

4.0.16 Visualisation of clustering

In assessing the validity of clustered space, we require a level of exploratory data analysis. To reveal features of interest, we plot the reduced 2D dataset and apply interactivity coupled with a selection of visualisation techniques described below. This section outlines the different visualisation methods which are used.

4.0.17 Viewing The 2D Species Embeddings

Since the different DR algorithms return data on various scales, comparison between the outputs is not straightforward. To overcome this outputs in x and y are normalised (scaled between {0,1}), before being plotted as a scatterplot.

4.0.18 Exposing Overlapping Data

If the nodes within a tight-knit cluster overlap, this can cause obfuscate the results and limit the user's ability to select them. As an initial test, node sizes can be reduced. However, this may often result in points too small to pick. The other solution which was used is to create a force-directed graph where each point is strongly attracted to their initial position. Here we can apply collision detection, while still preserving the overall grouping of nodes within a cluster - a technique that was seen in **??**.

4.0.19 Gooey Effect (Gaussian Blur)

Taking a quote from Reinhardt [1975]: *"The more stuff in it, the busier the work of art, the worse it is. More is less. Less is more. "* and combining it with the work from **??**, we realise that showing each species, when observing overall clusters just add unnecessary clutter to the images. Instead, since we are only interested in the clusters as a unit, a 'gooey effect' filter can be applied. This works by merging nearby points into a single water-like blob using a gaussian blur[8]. Here since each point is allocated a colour, if a colour gradient exists, then there are multiple clusters occupying the same place. The aim of this is to reduce the cognitive load on the end-user by reducing the number of distinct objects that they need to take in.

4.0.20 Four Colours Theorem

When plotted, the number of clusters detected often exceeds the number of categorical colours available. In cartography, it has been noted that the colouring of neighbouring polygons should at most take four colours. This is the origin of the four colours theorem Appel and Haken [1976], of which a greedy implementation is applied.

The aim of this is to show item boundaries (for instance countries, or in our case clusters) while reducing ambiguity (if, say, two neighbours have the same colour). The algorithm I adapted uses the Delaunay tesselation scripts contained within DataDrivenDocuments.js (d3js) Bostock [2012]. This partitions our plane into polygon-regions, each of which includes boundaries at the furthest distance

[8]Here a gaussian blur of standard deviation 3.7 and a colour matrix [1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 37 -5] is used.

from each point (Voronoi cells) Watson [1981]. First, we chose a random cell and assign it a colour. Next, all its neighbours are recursively iterated, giving them the lowest possible colour in a list, which does not match any of their neighbours. Although such a greedy approach does not produce an optimum result, it allows for the colouring of data with ≤ 5 distinct colours, as is shown in Figure 4.13.

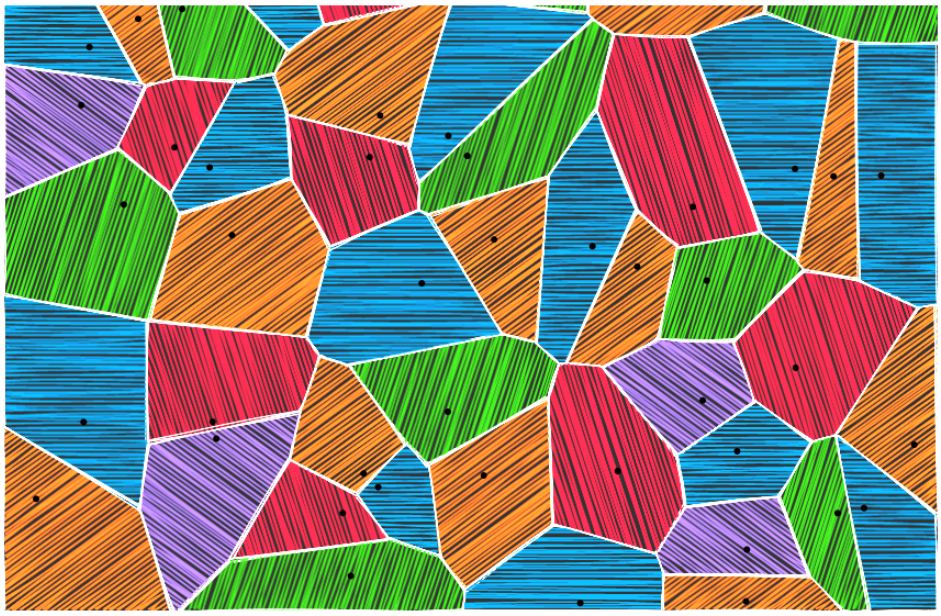


Figure 4.13: **An example 4 colour matching** This uses the first implementation of the algorithm mentioned in Subsection 4.0.20. The greedy approach does not often find the optimum solution, which may result in 5 colours instead. Observable Notebook : Daniel Ellis [2019]

Having defined all the visualisation techniques we move on to explain the clustering algorithms which are used, and how 'goodness of fit' may be measured in the clustering context.

4.0.21 Cluster Evaluation

The previous section discussed methods of visualising the reduced data for use with interactive exploratory data analysis. In this section we look at the use of vector clustering algorithms[9] (Subsection 4.0.22) to highlight groups in a 2D dataset, as well an automated method of assessing the quality of the clusters selected (Subsubsection 4.0.22.1) and feature extraction (Subsection 4.0.23).

4.0.22 Automated Selection Of Clusters

When it comes to clustering data points in a dataset, there exist a range of methods which may accomplish a task, Figure 4.14. Most often, the k-means [MacQueen, 1967], is used as it is fast and straightforward to understand. However, its linear method of partitioning cannot capture the splits

[9]Vector clustering is the grouping of data based on their proximity or density to other nearby points

between non-linear relationships of real data. The other problem is that an estimate for the number of expected clusters is required - something that is often unknown without prior understanding of the data. When this is the case, often it is easier to select the nodes with interactivity manually.

In contrast, density-based clustering techniques such as GMM ([Pedregosa et al., 2011a]) or DBSCAN ([Ester et al., 1996]) tend to be better at locating non-linear trends in the data. The DBSCAN algorithm asses the distribution of data across a specific location. This allows clusters with a high density of datapoints to be located without the need for a predefined number as an input. Another method: OPTICS (Ordering Points To Identify the Clustering Structure) [Ankerst et al., 1999], shall be used[10]. This is an adaptation of the DBSCAN algorithm which does not require the specification of a minimum distance between points (for the density estimate)- instead, we specify a gradient for the distribution and the minimum number of points for a cluster to be classified.

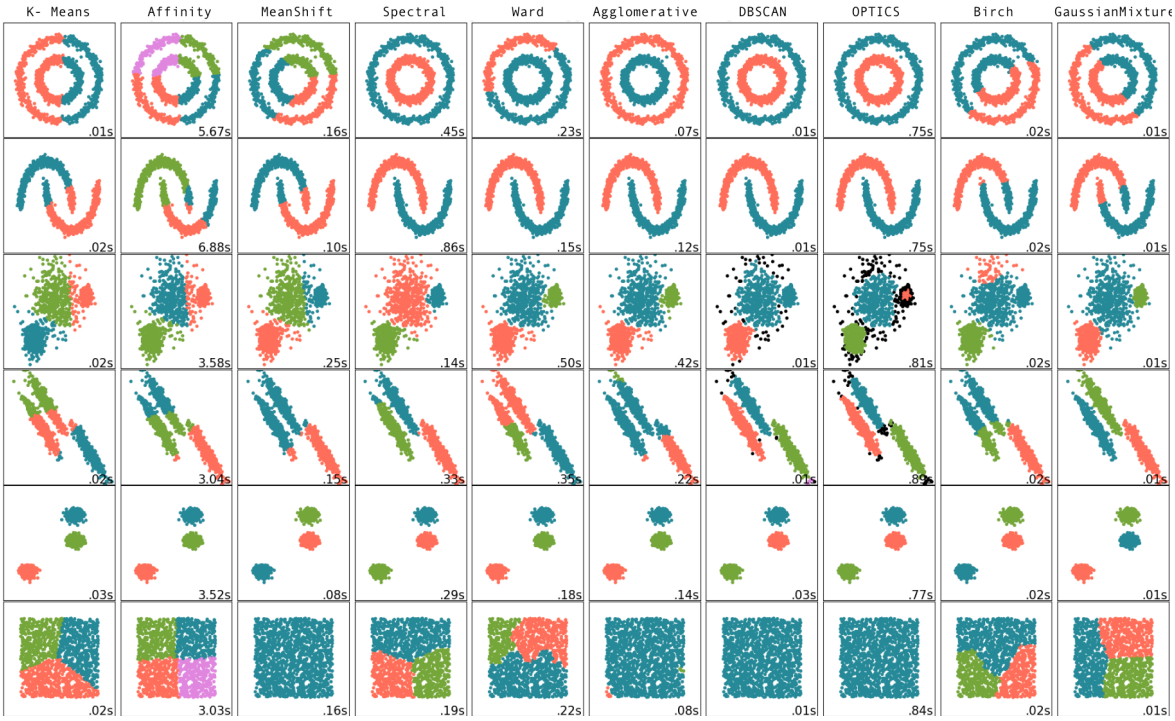


Figure 4.14: **A comparison of different clustering methods on a toy dataset.** The plot shows the performance of several vector clustering algorithms in Scikit-Learn. Cluster algorithms are represented across the horizontal axis and several types of datasets are across the vertical. Clustered groups are coloured. Source: sklearn [2019]

When deciding which algorithms to use, each algorithms' ability to partition non-linear data is considered. The first two rows of Figure 4.14 show data which cannot be partitioned linearly, here spectral, DBSCAN and optics are the only clustering algorithms to identify both correctly. It is for this reason that we shall look at these for the remainder of the chapter.

[10]If using Python 2, the library for this needs to be extracted from the sci-kit-learn library for python3 package and altered to run with the previous version. (See copy in attached code.)

In selecting a value for the results section, several clustering algorithms, with a wide range of input parameters, are run. From these, the simulation with the best silhouette coefficient (Subsubsection 4.0.22.1) is taken.

4.0.22.1 Clustering (Silhouette) Coefficient

The silhouette measure is a tool used for acessing the validity of a set of clusters. Here each cluster is represented as a silhouette, based on the comparison of its tightness and separation. To calculate the silhoette coefficient we look at the intra-cluster a and the mean inter-cluster[11] distance b. The silhouette cluster can then be described using **??**:

$$s(i) = \frac{b(i) - a(i)}{\max a(i), b(i)} \tag{4.6}$$

This gives a value $-1 \leq s(i) \leq 1$. Values near zero suggest overlapping clusters, 1 - dense, well-separated clusters and negative values indicate that a sample may have been incorrectly classified. In using this method, we can get an overview of how well individual objects lie within their assigned cluster.

4.0.23 Feature Extraction

Upon establishing a set of DR datasets, and their groups (the clusters of species they contain), it is important to evaluate what input features they represent. Rather than doing this manually we make use of Random Forests - described below.

4.0.23.1 Random Forrests

Random forests [Breiman, 2001], are a subset of ML algorithms called ensemble learning. This means that they train a large number of decision trees, each on a random subset of the original features. A decision tree is a tree formed from a series of conditionals[12], much like a perceptron network (Subsubsection 2.0.26.2) with binary activation functions. Random forests introduce a level of additional randomness by selecting only a subset on which to create each decision tree. This may introduce a higher bias, but lowers the overall model variance, which creates a better (more robust) model. Such methods have been applied to replacing the computationally expensive process of chemistry integration of GEOS-Chem (a global 3D model of tropospheric chemistry) [Keller and Evans, 2019] and the

[11] Inside and between different clusters.

[12] Questions with a True/False answer

prediction of global sea-surface iodine based on observations coupled with sea-surface temperature, depth, and salinity [Sherwen et al., 2019].

4.0.23.2 Calculating Importance Using Random Forrests

Since random forests are in essence a collection of decision trees, it is possible to generate a 'decision tree aggregate' to visualise the ensemble structure of the random forest [Ellis and Sherwen, 2019] (Figure 4.15). Alternatively, if all that is required is the relative importance of each feature, the RandomForrestClassifier from Pedregosa et al. [2011b] provides a quick and easy way of understanding which features matter, [Géron, 2017]. This works by aggregating the weighted nodes which use a certain feature using the number of samples and then scales the result to 1. We use this method to access the overall importance of features within each DR output and identify the differences between clusters.

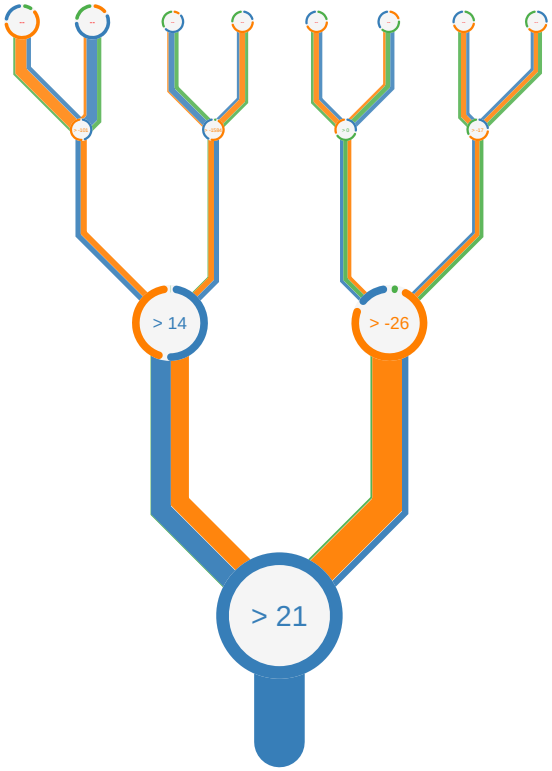


Figure 4.15: **A decision tree aggrogate from a random forrest plotted with the Epiphyte version of the TreeSurgeon program [Ellis and Sherwen, 2019].** The data originates from Sherwen et al. [2019] and the imporance of Tempearature (blue), Depth (orange) and Chlorophyll *a* (green).

NOTE: The only downside is that Random Forrests are in themselves ML techniques which also need to be evaluated. To do this, as they are simply being used as indicators of cluster properties which we are to explore further, we can initiate a collection of 300 random Forrest classifiers, from which we

take the median. A sort of ensemble learning from an ensemble.

4.0.24 Results

There exist many methods to define the chemical structure of the species within the MCM. In this section, we attempt to evaluate their effectiveness for exhibiting the defining functional groups and characteristics used for constructing the mechanism. First, we explore the distribution of clusters and the ability of different DR algorithms to visually separate various groups of chemistry (Subsection 4.0.25). Next, the functional groups (taken from the MCM development protocol) are explored within each DR algorithm (Subsection 4.0.26). Finally, a selected example for each DR method is taken and explored in further detail (Subsection 4.0.27).

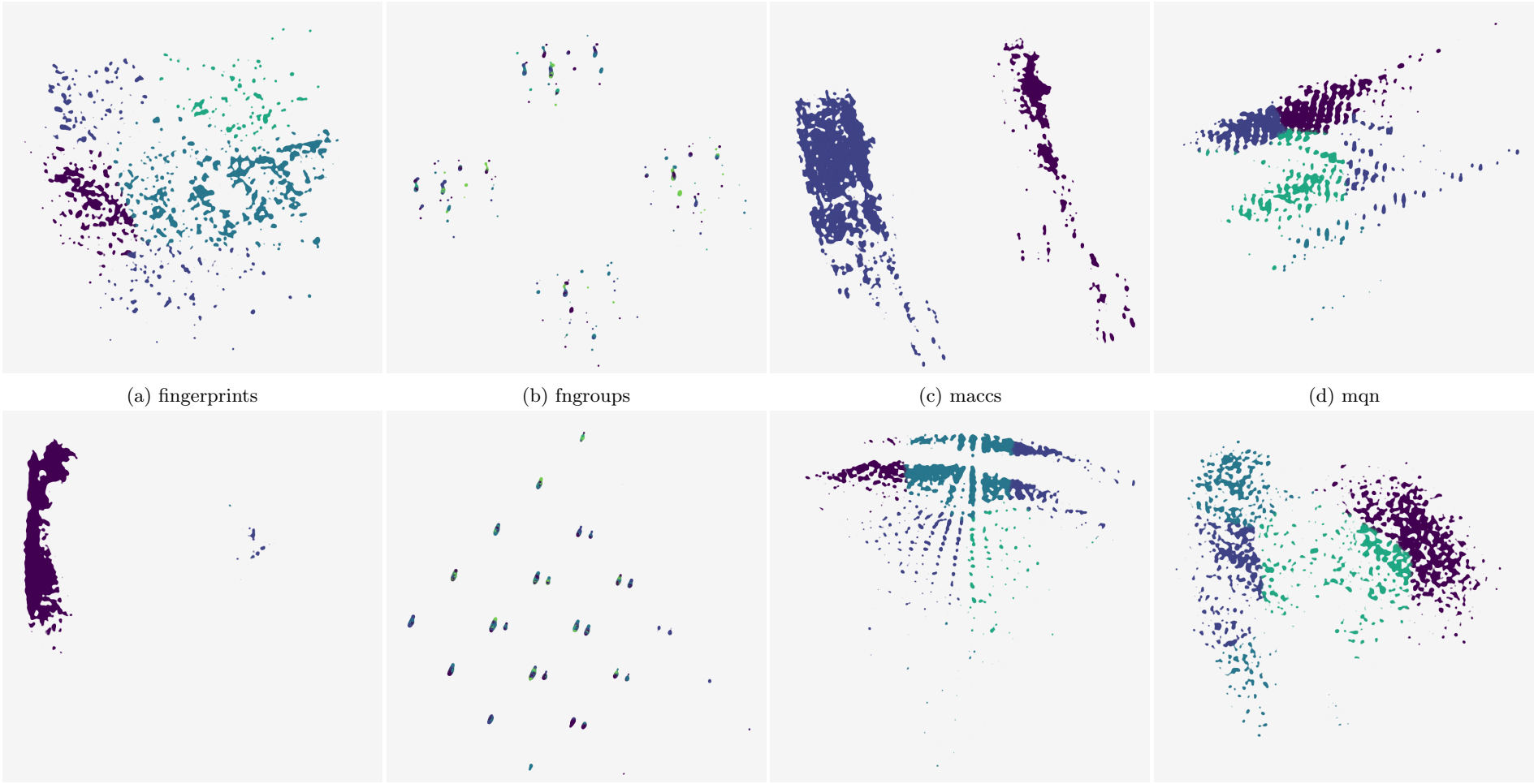
4.0.25 Cluster Distribution

Start with the visual comparison and compare it with the silhouette values.

Principle Component Analysis

DR	input	silhouette	groups
PCA	fngroups	0.9122	141
PCA	protocol	0.8761	149
PCA	node2vec	0.8569	3
PCA	maccs	0.6563	2
PCA	mqn	0.4041	8
PCA	smiles	0.3648	6
PCA	fingerprints	0.3529	6
PCA	spec	0.3364	6

Table 4.2: The inputs to the PCA dimensionality reduction algorithm sorted by the best obtained silhoette coefficient.



(a) fingerprints (b) fngroups (c) maccs (d) mqn

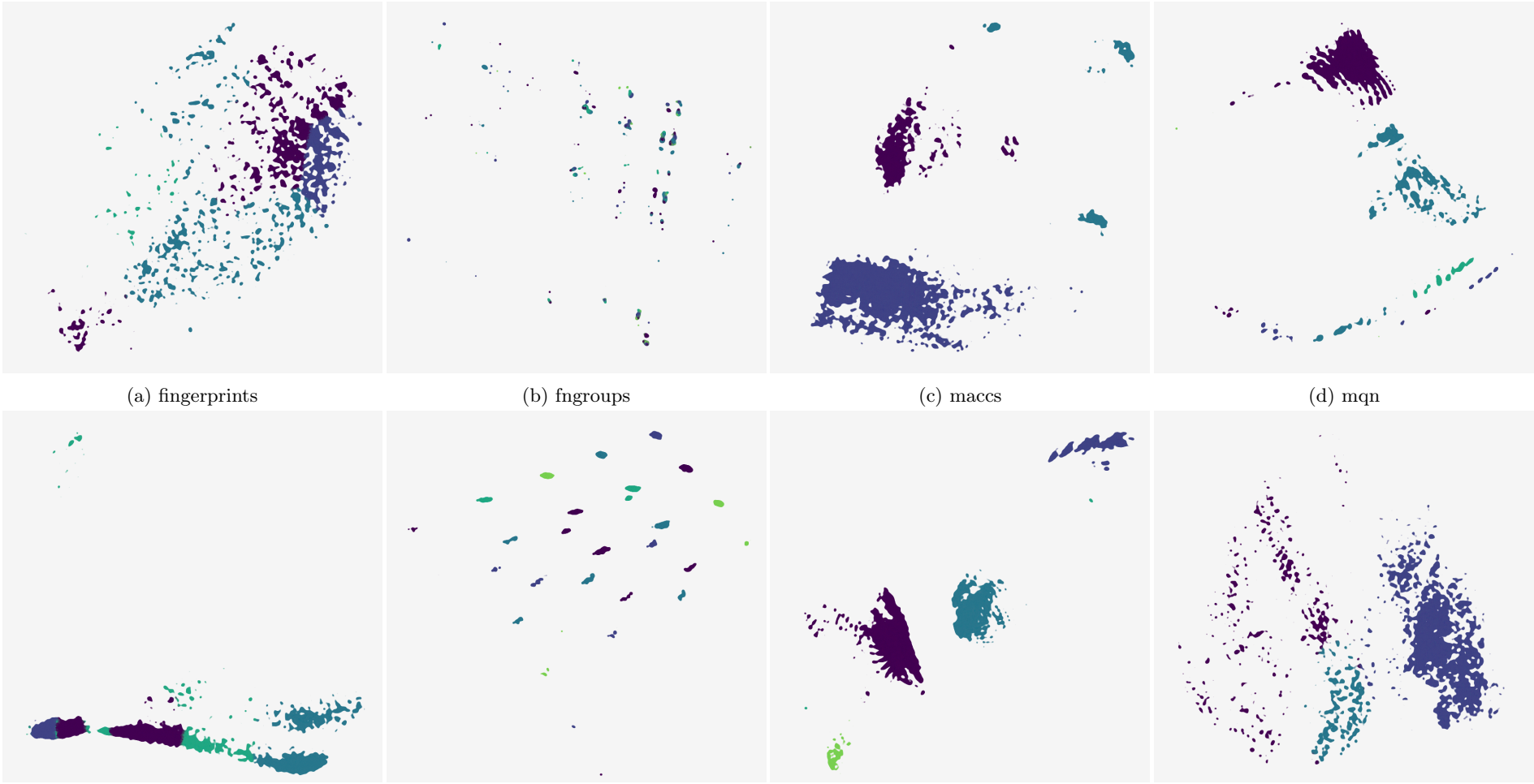
(e) node2vec (f) protocol (g) smiles (h) spec

Figure 4.16: **Comparing clusters for all inputs after a reduction to 2 dimensions using Principle Component analysis.** Each graph has undergone several clustering algorithms under a range of parameters. The result with the best silhouette coefficient has been chosen. Colours follow the greedy four colour theorem and are there only to indicate the contrast between cluster boundaries.

Auto Encoder Encoding

DR	input	silhouette	groups
AE	fngroups	0.9249	140
AE	protocol	0.8992	27
AE	smiles	0.6897	5
AE	mqn	0.6572	12
AE	maccs	0.6241	3
AE	node2vec	0.5476	5
AE	spec	0.4238	3
AE	fingerprints	0.3189	8

Table 4.3: The inputs to the AutoEncoder dimensionality reduction algorithm sorted by the best obtained silhoette coefficient.



(a) fingerprints (b) fngroups (c) maccs (d) mqn

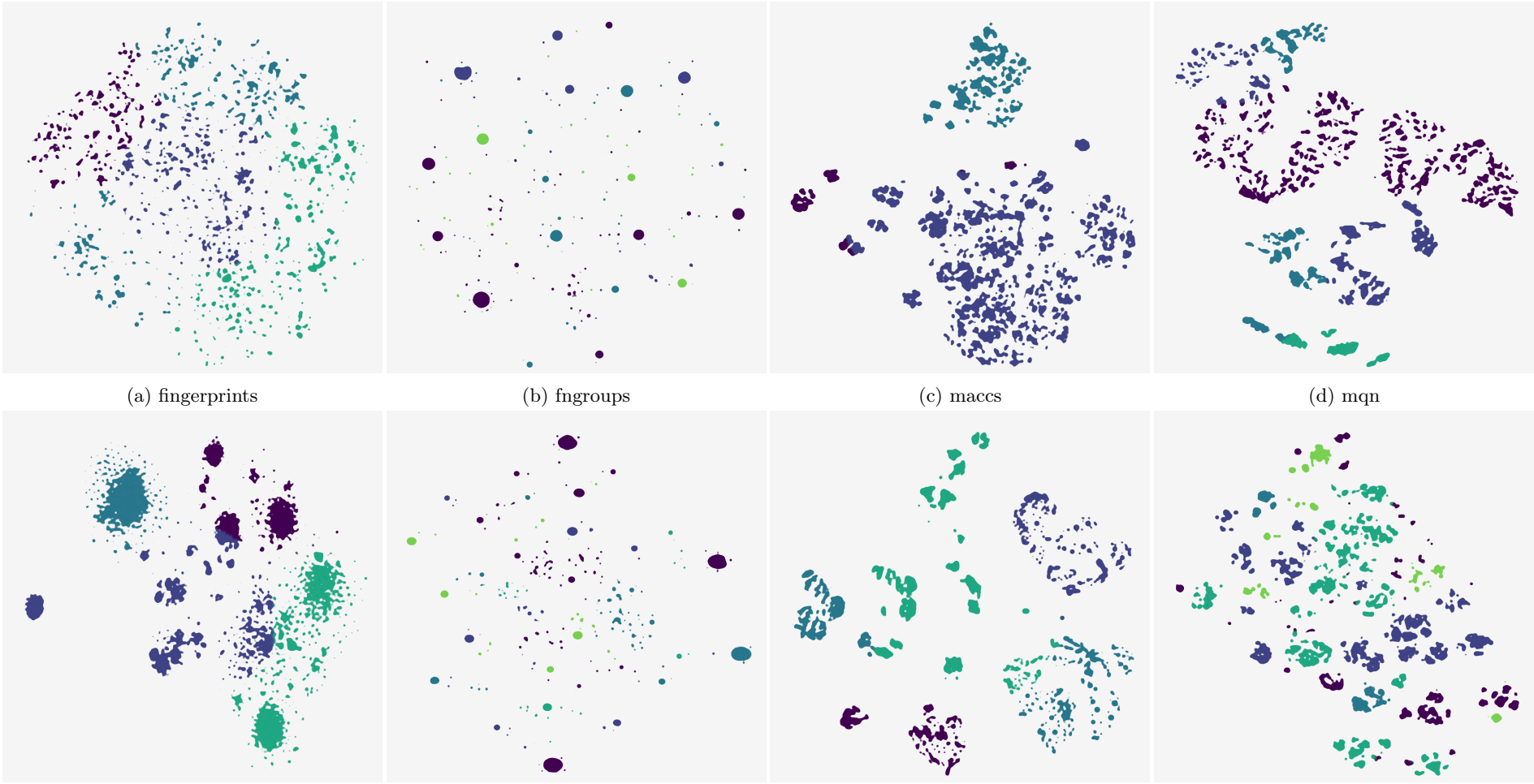
(e) node2vec (f) protocol (g) smiles (h) spec

Figure 4.17: **Comparing clusters for all inputs after a reduction to 2 dimensions using an AutoEncoder.** Each graph has undergone several clustering algorithms under a range of parameters. The result with the best silhouette coefficient has been chosen. Colours follow the greedy four colour theorem and are there only to indicate the contrast between cluster boundaries.

t-Distributed Stochastic Neighbor Embedding

DR	input	silhouette	groups
t-SNE	fngroups	0.7458	106
t-SNE	protocol	0.5688	51
t-SNE	smiles	0.4808	6
t-SNE	node2vec	0.4359	6
t-SNE	maccs	0.4295	3
t-SNE	spec	0.3781	35
t-SNE	mqn	0.3684	8
t-SNE	fingerprints	0.3539	6

Table 4.4: The inputs to the t-SNE dimensionality reduction algorithm sorted by the best obtained silhoette coefficient.

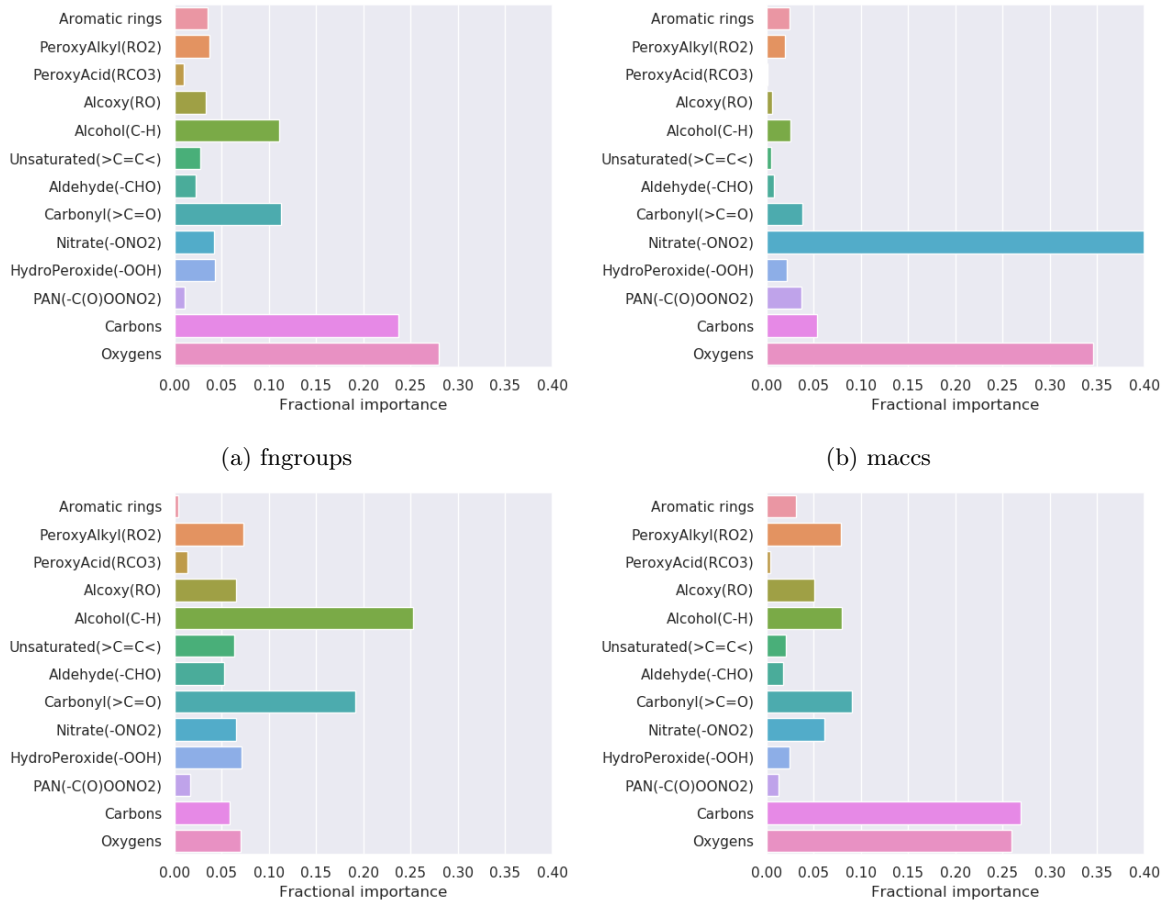


(a) fingerprints (b) fngroups (c) maccs (d) mqn

(e) node2vec (f) protocol (g) smiles (h) spec

Figure 4.18: **Comparing clusters for all inputs after a reduction to 2 dimensions using t-SNE.** Each graph has undergone several clustering algorithms under a range of parameters. The result with the best silhouette coefficient has been chosen. Colours follow the greedy four colour theorem and are there only to indicate the contrast between cluster boundaries.

4.0.26 Feature Selection Comparison



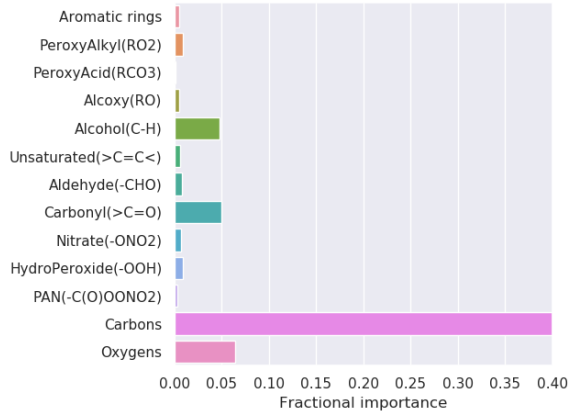
(a) fngroups

(b) maccs

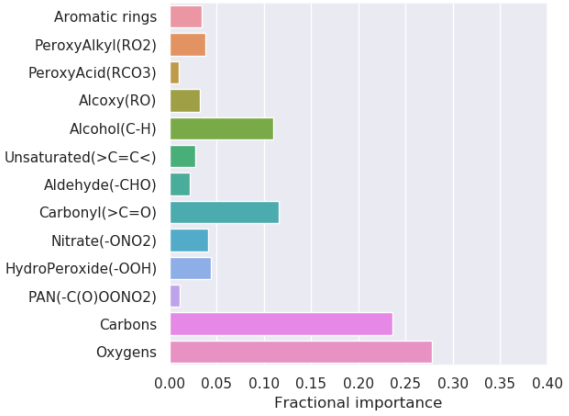
(c) protocol

(d) smiles

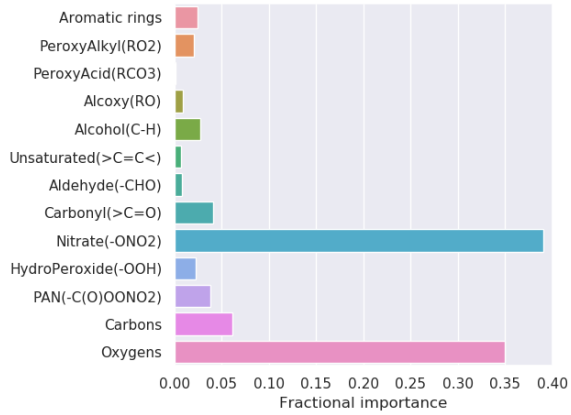
Figure 4.19: **Comparing feature importance for PCA clusters.**



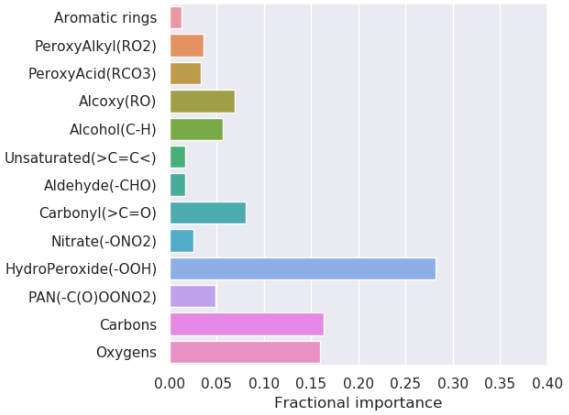
(a) fingerprints



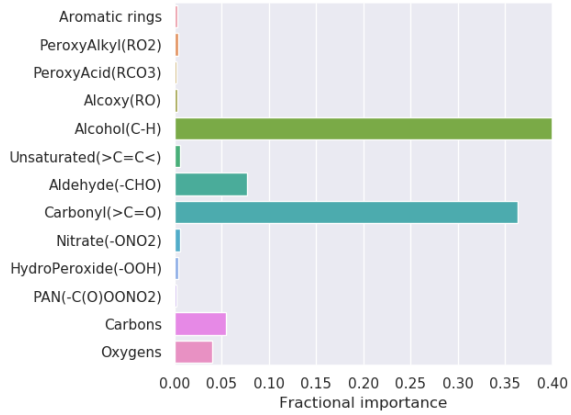
(b) fngroups



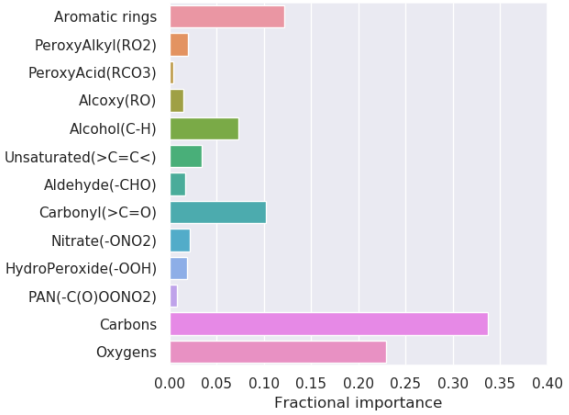
(c) maccs



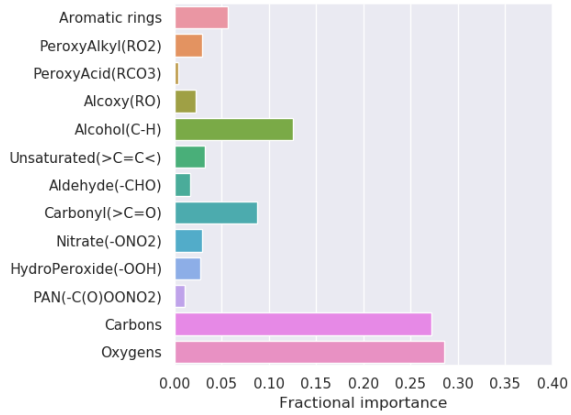
(d) node2vec



(e) protocol

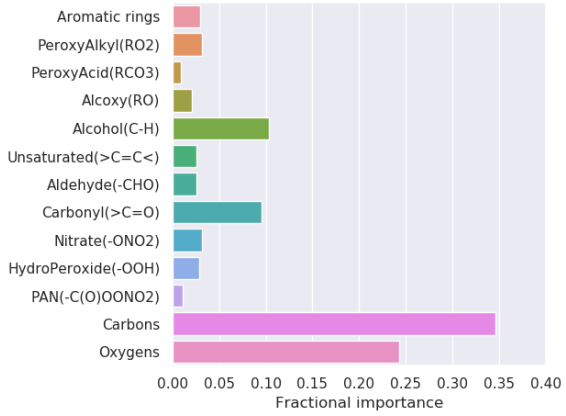


(f) smiles

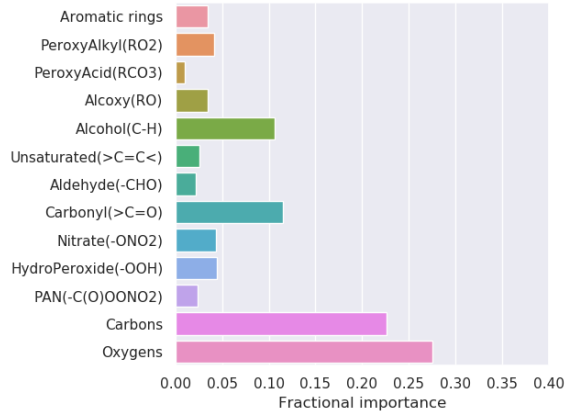


(g) spec

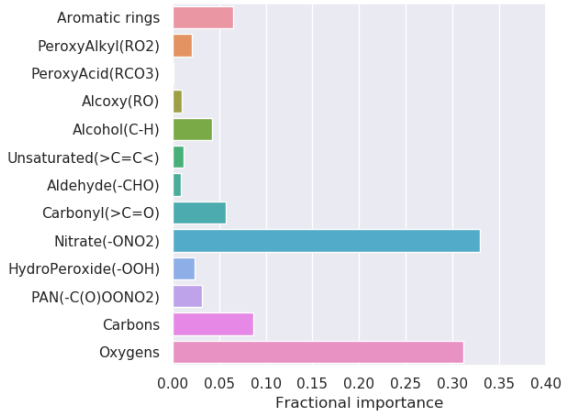
Figure 4.20: **Comparing feature importance for AE clusters.**



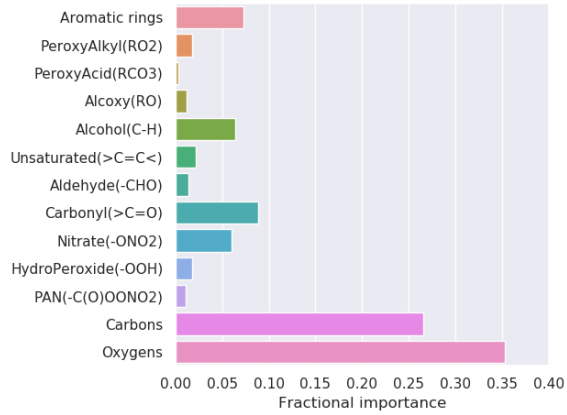
(a) fingerprints



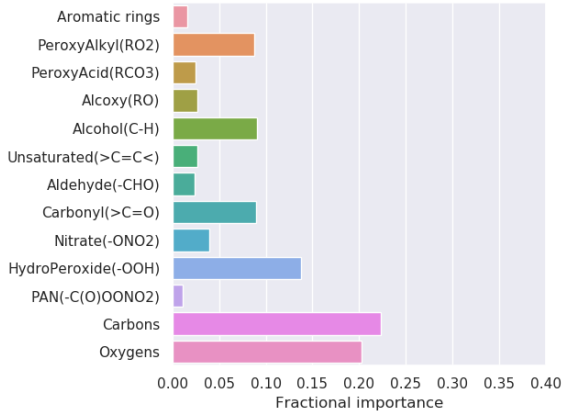
(b) fngroups



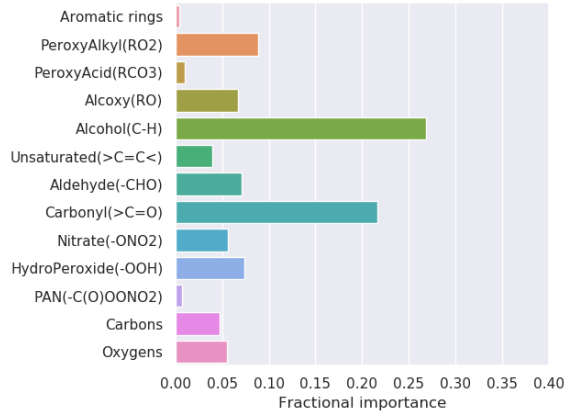
(c) maccs



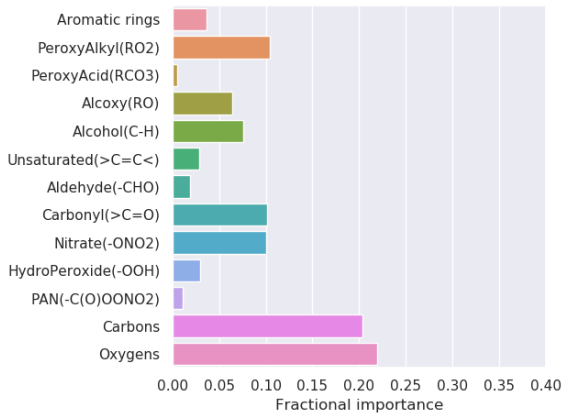
(d) mqn



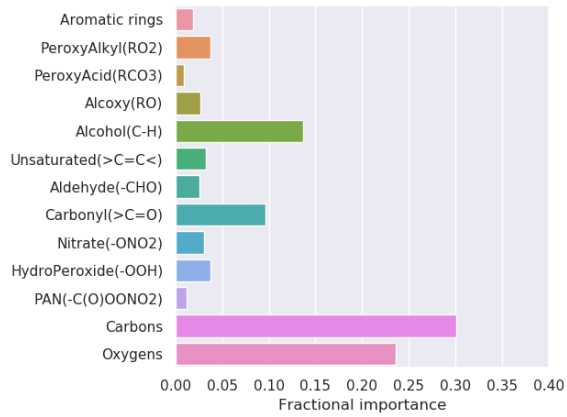
(e) node2vec



(f) protocol



(g) smiles



(h) spec

Figure 4.21: **Comparing feature importance for t-SNE clusters.**

4.0.27 Individual Cluster Comparison

4.0.28 Conclusions

DR can be used to find patterns in dataset which is best

interaction

There are a range of inputs each showing a few different things.

Depending on what properties we are intereste dwe may select accordingly

Bibliography

Alon, U., Zilberstein, M., Levy, O., and Yahav, E. (2019). Code2Vec: Learning Distributed Representations Of Code. http://dl.acm.org/citation.cfm?doid=3302515.3290353.

Anderson, C. (2008). The End Of Theory: The Data Deluge Makes The Scientific Method Obsolete. *online*. http://www.wired.com/print/science/discoveries/magazine/16-0.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60. https://doi.org/10.1145/304181.304187.

Appel, K. and Haken, W. (1976). Every planar map is four colorable. *Bull. Amer. Math. Soc.*, 82(5):711–712. https://projecteuclid.org:443/euclid.bams/1183538218.

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(9):2497–2517. https://www.atmos-chem-phys.net/5/2497/2005/.

Baillargeon, R. and Carey, S. (2012). Core cognition and beyond: The acquisition of physical and numerical knowledge. *Early childhood development and later outcome*.

Bostock, M. (2012). D3.js - data-driven documents. http://d3js.org/.

Box, G. E. P. (1976). Science And Statistics. *Journal of the American Statistical Association*, 71(356):791–799. https://www.tandfonline.com/doi/abs/10.1080/01621459.1976.10480949.

Breiman, L. (2001). Random Forests. *Machine learning*, 45(1):5–32. https://doi.org/10.1023/A:1010933404324.

Cohen, E. (2018). Node2Vec: Embeddings For Graph Data. https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef.

Daniel Ellis (2019). D3-Fourcolour Voronoi. https://observablehq.com/@wolfiex/d3-fourcolour-voronoi.

Dataman (2019). Convolutional Autoencoders For Image Noise Reduction. https://towardsdatascience.com/convolutional-autoencoders-for-image-noise-reduction-32fce9fc1763.

Descartes, R. and Lafleur, L. J. (1960). *Meditations On First Philosophy*. Bobbs-Merrill New York. http://selfpace.uconn.edu/class/percep/DescartesMeditations.pdf.

Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., and Zhi, D. (2019). Gene2Vec: Distributed Representation Of Genes Based On Co-Expression. *BMC genomics*, 20(Suppl 1):82. http://dx.doi.org/10.1186/s12864-018-5370-x.

Durant, J. L., Leland, B. A., Henry, D. R., and Nourse, J. G. (2002). Reoptimization Of Mdl Keys For Use In Drug Discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280. https://www.ncbi.nlm.nih.gov/pubmed/12444722.

Ellis, D. (2019). Chemical Kinetic Interactions Cover Image. https://s100.copyright.com/AppDispatchServlet?startPage=i&publisherName=Wiley&publication=kin&contentID=10.1002%2Fkin.21180&endPage=i&title=Cover+Image%2C+Volume+50%2C+Issue+6.

Ellis, D. and Sherwen, T. (2019). Wolfiex/treesurgeon: Wollemia. https://doi.org/10.5281/zenodo.3346817.

Ester, M., peter Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press.

F.R.S., K. P. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. https://doi.org/10.1080/14786440109462720.

Géron, A. (2017). *Hands-On Machine Learning With Scikit-Learn And Tensorflow: Concepts, Tools, And Techniques To Build Intelligent Systems*. O'Reilly Media. https://books.google.co.uk/books?id=khpYDgAAQBAJ.

Grover, A. and Leskovec, J. (2019). Node2vec: Scalable feature learning for networks. Accessed: 2019-10-21.

Hamadache, M. and Lee, D. (2017). Principal Component Analysis Based Signal-To-Noise Ratio Improvement For Inchoate Faulty Signals: Application To Ball Bearing Fault Detection. *International journal of control, automation, and systems*, 15(2):506–517. https://doi.org/10.1007/s12555-015-0196-7.

Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D., and Pletnev (2013). Inchi - The Worldwide Chemical Structure Identifier Standard. *Journal of cheminformatics*, 5(1):7. http://dx.doi.org/10.1186/1758-2946-5-7.

Hernandez, W. and Mendez, A. (2018). Application Of Principal Component Analysis To Image Compression. In Göksel, T., editor, *Statistics - Growing Data Sets and Growing Demand for Statistics*. InTech. http://www.intechopen.

com/books/statistics-growing-data-sets-and-growing-demand-for-statistics/application-of-principal-component-analysis-to-image-compression.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441. https://doi.org/10.1037%2Fh0071325.

Jean, N., Wang, S., Samar, A., Azzari, G., Lobell, D., and Ermon, S. (2018). Tile2Vec: Unsupervised Representation Learning For Spatially Distributed Data. http://arxiv.org/abs/1805.02855.

Jolliffe, I. T. and Cadima, J. (2016). Principal Component Analysis: A Review And Recent Developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 374(2065):20150202. http://dx.doi.org/10.1098/rsta.2015.0202.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). Scipy: Open source scientific tools for Python. http://www.scipy.org/.

Keller, C. A. and Evans, M. J. (2019). Application of random forest regression to the calculation of gas-phase chemistry within the geos-chem chemistry model v10. *Geoscientific Model Development*, 12(3):1209–1225. https://www.geosci-model-dev.net/12/1209/2019/.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet Classification With Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Landrum, G., Tosco, P., Kelley, B., sriniker, gedeck, NadineSchneider, Vianello, R., Dalke, A., Cole, B., AlexanderSavelyev, Turk, S., Ric, Swain, M., Vaucher, A., N, D., Wójcikowski, M., Pahl, A., JP, strets123, JLVarjo, O'Boyle, N., Berenger, F., Fuller, P., Jensen, J. H., Sforna, G., DoliathGavid, Cosgrove, D., Nowotka, M., Leswing, K., and van Santen, J. (2019). Rdkit 2019-03-2 (q1 2019) release. https://doi.org/10.5281/zenodo.2864247.

Leite, N. M. N., Pereira, E. T., Gurjão, E. C., and Veloso, L. R. (2018). Deep convolutional autoencoder for eeg noise filtering. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2605–2612.

Lynch, H. (2011). *Infant Places, Spaces And Objects: Exploring The Physical In Learning Environments For Infants Under Two*. PhD thesis. http://dx.doi.org/10.21427/D73W37.

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

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press. https://projecteuclid.org/euclid.bsmsp/1200512992.

(MDL), M. I. S. (1984). Maccs-ii.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation Of Word Representations In Vector Space. http://arxiv.org/abs/1301.3781.

Morozov, A. (2016). Modelling biological evolution: Linking mathematical theories with empirical realities. *Journal of Theoretical Biology*, 405:1 – 4. http://www.sciencedirect.com/science/article/pii/S0022519316301849.

Nguyen, K. T., Blum, L. C., van Deursen, R., and Reymond, J.-L. (2009). Classification Of Organic Molecules By Molecular Quantum Numbers. *ChemMedChem*, 4(11):1803–1805. http://dx.doi.org/10.1002/cmdc.200900317.

Noble, C. E. (1957). Human Trial-And-Error Learning. *Psychological reports*, 3(2):377–398. https://doi.org/10.2466/pr0.1957.3.h.377.

Oliphant, T. (2006). Guide to numpy.

Oliveira, B., Pereira, F., de AraÃ°jo, R., and Ramos, M. (2006). The hydrogen bond strength: New proposals to evaluate the intermolecular interaction using dft calculations and the aim theory. *Chemical Physics Letters*, 427(1):181 – 184. http://www.sciencedirect.com/science/article/pii/S000926140600861X.

Parsons, J., Holmes, J. B., Rojas, J. M., Tsai, J., and Strauss, C. E. M. (2005). Practical Conversion From Torsion Space To Cartesian Space For In Silico Protein Synthesis. *Journal of computational chemistry*, 26(10):1063–1068. http://dx.doi.org/10.1002/jcc.20237.

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. (2011a). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830. http://dl.acm.org/citation.cfm?id=1953048.2078195.

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. (2011b). Scikit-Learn: Machine Learning In Python . *Journal of Machine Learning Research*, 12:2825–2830.

People2Vec (2019). People2Vec. http://people2vec.org/.

Powell, V. (2020). Principal Component Analysis Explained Visually. http://setosa.io/ev/principal-component-analysis/.

Probst, D. and Reymond, J.-L. (2018). Smilesdrawer: Parsing And Drawing Smiles-Encoded Molecular Structures Using Client-Side Javascript. *Journal of chemical information and modeling*, 58(1):1–7. http://dx.doi.org/10.1021/acs.jcim.7b00425.

rdkit (2019). Rdkit. https://github.com/rdkit/rdkit/blob/24f1737839c9302489cadc473d8d9196ad9187b4/rdkit/Chem/MACCSkeys.py.

Reinhardt, A. (1975). *Art-As-Art: The Selected Writings Of Ad Reinhardt*. Documents of 20th-century art. Viking Press. https://books.google.co.uk/books?id=zyK4AAAAIAAJ.

Roberts, R. (1989). *Serendipity: Accidental Discoveries In Science*. Wiley Science Editions. Wiley. https://books.google.co.uk/books?id=hf57X0s4aPwC.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326. https://science.sciencemag.org/content/290/5500/2323.

Sherwen, T., Chance, R. J., Tinel, L., Ellis, D., Evans, M. J., and Carpenter, L. J. (2019). A machine-learning-based global sea-surface iodide distribution. *Earth System Science Data*, 11(3):1239–1262. https://www.earth-syst-sci-data.net/11/1239/2019/.

sklearn (2019). Comparing Different Clustering Algorithms On Toy Datasets — Scikit-Learn 0.21.3 Documentation. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html.

Spahn, V., Del Vecchio, G., Labuz, D., Rodriguez-Gaztelumendi, A., Massaly, N., Temp, J., Durmaz, V., Sabri, P., Reidelbach, M., Machelska, H., Weber, M., and Stein, C. (2017). A Nontoxic Pain Killer Designed By Modeling Of Pathological Receptor Conformations. *Science*, 355(6328):966–969. http://dx.doi.org/10.1126/science.aai8636.

T. Leube, B., Inglis, K., J. Carrington, E., and Sharp, P. (2018). Lithium transport in li 4.4 m 0.4 m Â 0.6 s 4 (m = al 3+ , ga 3+ and m Â= ge 4+ , sn 4+): Combined crystallographic, conductivity, solid state nmr and computational studies. *Chemistry of Materials*, 30.

Turanyi, T. and Tomlin, A. (2015). *Analysis Of Kinetic Reaction Mechanisms*. Springer. http://eprints.whiterose.ac.uk/84294/.

Wang, S.-G. and Schwarz, W. H. E. (2009). Icon Of Chemistry: The Periodic System Of Chemical Elements In The New Century. *Angewandte Chemie*, 48(19):3404–3415. http://dx.doi.org/10.1002/anie.200800827.

Watson, D. F. (1981). Computing The N-Dimensional Delaunay Tessellation With Application To Voronoi Polytopes*. *The Computer Journal*, 24(2):167–172. https://doi.org/10.1093/comjnl/24.2.167.

Weininger, D. (1988). Smiles, A Chemical Language And Information System. 1. Introduction To Methodology And Encoding Rules. *Journal of chemical information and computer sciences*, 28(1):31–36. https://pubs.acs.org/doi/abs/10.1021/ci00057a005.

Yu-ChenLo (2018). Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today*, 23(8):1538 – 1546. http://www.sciencedirect.com/science/article/pii/S1359644617304695.