TRANSFORMING HEALTHCARE DATA WITH GRAPH-BASED TECHNIQUES USING SAIL DATABANK (JAN - MAY 2023)

Zoe Hancox
University of Leeds
School of Computer Science
ll15zlh@leeds.ac.uk

Dan Schofield
NHS England
Transformation Directorate
daniel.schofield1@nhs.net

ABSTRACT

This report presents an approach to analysing disease set patterns using hypergraphs and multimorbidity data. Hypergraphs provide a powerful framework for modelling complex relationships among diseases, and their integration with multimorbidity data offers a comprehensive understanding of the co-occurrence of multiple diseases within patient populations. Additionally, this work extends on the previous work by incorporating mortality information into the hypergraphs and exploring the concept of temporality as hyperarc weights. The inclusion of mortality data enhances the analysis by considering the impact of diseases on patient outcomes. Whilst temporality enables the inclusion of irregular time intervals which captures the dynamic nature of multimorbidity patterns over time.

To facilitate the understanding of hypergraphs and their applications in the multimorbidity domain, an interactive applet has been developed. This serves as an educational tool and visualisation device, teaching users about undirected and directed hypergraphs and demonstrating their usefulness in analysing complex disease relationships. We hope that our applet and the code bases we have created will promote the dissemination of knowledge about hypergraphs and their applications, empowering individuals to explore and comprehend complex healthcare data in the multimorbidity domain.

1 Introduction

Multimorbidity is defined as having 2 or more long term chronic health conditions occurring simultaneously. Multimorbidity is associated with increased health service utilisation [1].

People are living longer and so the population is ageing. This is due to improvements in lifestyle through things such as diet, not smoking and exercise. Treatment improvements mean that more people are surviving acute conditions. The number of hospitals and other healthcare services are not increasing fast enough to keep up with this rising ageing population and increased health service utilisation required by patients with multimorbidity, so healthcare services are being burdened.

Approximately 1 in 4 patients in primary care within the UK had multiple chronic conditions in 2018 [1]. Unfortunately these multimoribidities reduce life quality and can increase mortality. Even more concerning is that in 2015 54% of people over 65 years old had multimorbidity, and in just 20 years this is expected to rise to 67.8% [2].

One 2012 study using data from around 14 million patients found that hypertension, depression/anxiety followed by chronic pain were the most prevalent multimorbidity conditions. They also found than females (30%) were more likely to have multimorbidity than males (24.4%). Those with a low socioeconomic status had 4.2% more people with multimorbidity than the highest socioeconomic status group. Additionally, around half of General Practitioner (GP) consultations and hospital admissions were for multimorbidity. 78.7% of prescriptions were for multimorbidity [1].

As multimorbidity becomes more prevalent it is important that research in multimorbidity develops so that new strategies can be approved to change health frameworks and policies, to prevent accumulation of conditions, better manage those with multiple condition and reduce the burden on healthcare.

This project had various overarching objectives which we aimed to achieve. Mainly we wanted to see if we could find salient patterns in multimorbidity pathways and multimorbidity disease sets which could be used to:

1. Identify contributory relationships between diseases.
2. Identify important multimorbidity sets within different population demographics.
3. Prevent multimorbidity development.
4. Help prioritise care services.
5. Optimise treatment and management to improve pain and reduce healthcare costs.

Using hypergraphs in prediction models enables more complexities and relationships from the data to be included. Where a standard graph $\mathcal{G}(v, e)$ can only connect 2 objects/nodes with one edge (pairwise connections), hypergraphs $\mathcal{H}(v, e)$ can connect 2 or more nodes per edge. Hypergraphs enable multimorbidity edge connections between diseases to be established rather than only co-morbidity. Where the nodes are the diseases and the hyperedges are the multimorbidity sets.

This work has two accompanying repositories under the names **hypergraph-mm** and **hypergraphical**, these are available on GitHub at `https://github.com/nhsx/hypergraph-mm` and `https://github.com/nhsx/hypergraphical/tree/main` respectively which contains various reproduce the results contained below.

2 Background

The prior project [3] built upon the work in [4] which used hypergraphs for population-based multimorbidity representation and analysis. The outcomes included: mortality, prevalence, and resource utilisation were considered previously. But with a focus on prevalence, considering what successors and predecessors are likely to occur within the population. Explicitly with the prevalence being used to weight the hyperedges.

Various hyperedge weighting systems have been explored to experiment with contribution towards progressive diseases. Two main formulae were experimented with: the overlap coefficient, Sørensen-Dice coefficient (the modified power set and the modified complete set versions). How individuals contribute to the graph by using power sets, exclusive/single contribution, or ordered progression were explored. For example, if an individual had an ordered disease set of $\{A, B, C\}$ then the ordered progression set would be $\{A, B\}, \{A, B, C\}$. From this, progression was used as the contribution using the modified Sørensen-Dice coefficient (complete set).

[4] used eigenvector centrality within undirected hypergraphs to measure the influence diseases have on each other and their importance that single set diseases have in the original hypergraph representation. The dual hypergraph representation was used to measure the influence and importance for multimorbidity disease sets. Directionality was later introduced within these hypergraph models to observe sequential transitions between disease sets [3]. Work was done involving dual Hypergraphs (switching the roles of the nodes and edges, so that the nodes are disease sets and the edges are the individual diseases linking the disease sets). Dual hypergraphs were successful for undirected hypergraph dual representations, but information loss occurred in the directed hypergraphs.

Analysis on hypergraph structures were carried out using: Random Walks, PageRank, successor/predecessor detection (for directional relationships between disease progression pairs, the strength from incoming and outgoing connections) and Eigenvector Centrality (to rank important diseases that contribute to other diseases).

Stratification based on demographics including age, sex and deprivation status (how relationships between morbidities differ vary between demographics) was undertaken. Previous work also started exploring how incorporating mortality into these hypergraphs could be used to demonstrate importance of progressions to mortality [3].

2.1 Graphs and Hypergraphs - Constructing Incidence and Adjacency Matrices

Figure 1 shows a visualisation of the different types of graphs we discuss in this report.

Here we will defined a graph as a standard graph, that is a graph with nodes v and edges e such that $\mathcal{G}(v, e)$. The edges in a standard graph may connect only two nodes (Undirected standard graph Figure 1a, directed standard graph Figure 1b).

Hypergraphs were defined by Berge [5], as an extension of graphs which enable relationships between many nodes, rather than just pairwise relationships. Both graphs and hypergraphs can be undirected (Figure 1c) or directed (Figure 1d-f). Directed hypergraph edges are called hyperarcs. In a hyperarc the tail nodes are the node(s) from which the hyperarc begins and the head node(s) are where the hyperarc terminates.

There are three main different types of directed hypergraphs: F-Hypergraphs (Figure 1e), B-Hypergraphs (Figure 1d) and BF-Hypergraphs (Figure 1f). F-Hypergraphs can have an infinite amount of head nodes in one hyperarc but only

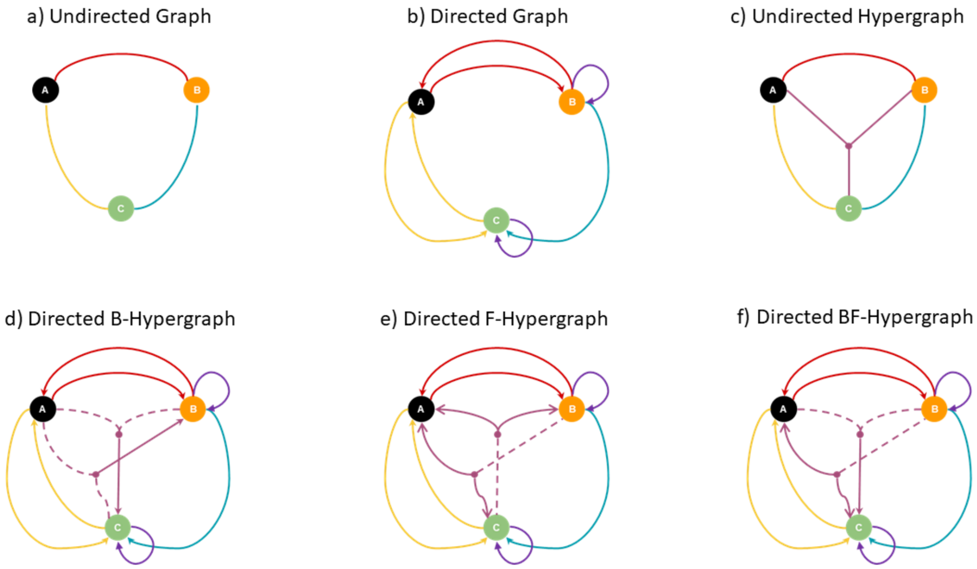

Figure 1: The different types of graphs (standard and hypergraphs) discussed in this report.

one tail node. B-hypergraphs are the opposite of F-Hypergraphs, in a B-Hypergraph each hyperarc can have only one head node but may have an infinite number of tail nodes. BF-Hypergraphs are the most computationally complex, these hyperarcs have no restraints on the number of head or tail nodes.

We consider B-hypergraphs as a simple starting point. Noting the limitation that B-hypergraphs imply that no one ever recovers from a condition after their first recording of said condition.

2.1.1 Undirected Hypergraphs

Adjacency matrices are used to represent graphs in a matrix form. When constructing an adjacency matrix A for an undirected hypergraph, we first need to define an incidence matrix M where the rows are the nodes and the columns are the hyperedges $M = M(v, e)$.

We also need to calculate the diagonal matrices D_n and D_e. Where D_n is the diagonal matrix of the node degree $d(\cdot)$ and D_e is the diagonal matrix of the edge degree $\delta(\cdot)$. For nodes $u, v \in V$ we have that $D_n(u, v) = d(v)$ for $u = v$ and 0 otherwise. The diagonal matrix D_e is defined similarly such that for edges $e_i, e_j \in \mathcal{E}$, $D_e(e_i, e_j) = \delta(e_i)$ where $e_i = e_j$ and 0 otherwise.

Once the diagonal matrices are calculated the adjacency matrix can be calculated (Eqn 1. This tells us how many edges each of the nodes share with each other.

$$A_{ij} = MM^T - D_n \qquad A_{ij} = \begin{cases} k & \text{if node } i \text{ is connected to node } j \text{ through } k \text{ edges} \\ 0 & \text{if } i = j \text{ or if zero edges connect nodes } i \text{ and } j \end{cases} \tag{1}$$

Undirected graphs cannot have self-loops, as such the diagonal of A_{ij} will always be 0's $a_{uu} = 0$.

Eqn 1 calculates an unweighted adjacency matrix. But the weighted adjacency matrix can also be calculated, once the hyperedge weights are calculated, using the equation: $A_{ij} = MW_E M^T - D_n$ where W_E is the matrix of the weights of the hyperedge.

2.1.2 Directed B-Hypergraphs

The concatenation of two incidence matrices have been used to represent directed B-hypergraphs [3]. Such that the incidence is the connection of a node to an edge, either it's tail or head.

The incidence matrix M_D for a 3-node directed B-hypergraph $H_D(V,\mathcal{E})$, $V = \{D_1, D_2, D_3\}$ is defined such that when $M_D(i,j) = -1$, $D_j \in T(h_i)$, while when $M_D(i,j) = 1$, $D_j \in H(h_i)$. Note that self-looping hyperarcs cannot be explicitly represented using this representation as they are both the tail and head node of a hyperarc. Therefore, incidence matrix is split into its tail and head components M_D^- and M_D^+ respectively. After the tail M_D^- and head M_D^+ components have been defined, we can concatenate them to make a single incidence matrix M.

2.2 Hyperedge Weight Calculation

Two different types of Sørensen–Dice coefficients have been experimented with [3]. These were the Power Set Sørensen–Dice coefficient (raw prevalence count + power set in the denominator) and the complete set Sørensen–Dice coefficient (raw prevalence count + power set + super set in the denominator). The Sørensen–Dice coefficient (complete set) was used to calculate the hyperedge weights $W(e_i)$ as shown in Eqn 2.

$$W(e_i) = \frac{C(e_i)}{C(e_i) + \sum_{e_j \in \mathcal{P}(e_i)} w_j C(e_j) + \sum_{e_k \in \mathcal{S}(e_i)} w_k C(e_k)}, \tag{2}$$

Where:

- $C(e_i)$ is the raw prevalence count for multimorbidity set e_i
- $\mathcal{P}(e_i)$ is the power set of hyperedges for multimorbidity set e_i (all subset disease sets)
- $\mathcal{S}(e_i)$ is the super set of hyperedges for multimorbidity set e_i (all disease sets containing e_i)
- w_j and w_i are optional weight coefficients for the penalisation terms. Assumed as unitary to not make presumptuous conclusions on the similarity of related disease sets

2.3 Directed Hyperarc Weight Calculation

The weighting of the hyperarcs considers disease prevalence by weighting its prevalence among other children and of its parent hyperedge, which overall is weighted by the prevalence of the hyperedge itself.

Let

$$\mathcal{K}(h_i) = \{h_j \, : \, p(h_j) = p(h_i)\} \tag{3}$$

be the set of all B-hyperarc children of the parent hyperedge $p(h_i)$. Where $\mathcal{K}(h_i)$ is the set of siblings for hyperarc h_i. And $W(p(h_i))$ is the hyperedge weight of the parent e_i. Then we define the weight for hyperarc h_i as

$$w(h_i) = W(p(h_i)) \frac{C(h_i)}{\sum_{h_j \in \mathcal{K}(h_i)} C(h_j)}. \tag{4}$$

2.4 Centrality Calculations

2.4.1 Eigenvector Centrality

Importance or centrality of a node is defined based on its connectivity to other nodes, and how many connections those connected nodes have to other nodes.

Eigenvector centrality was used for undirected graphs and is calculated by taking the weighted adjacency matrix and finding the Eigenvalues and respective Eigenvector.

2.4.2 Random Walks on Undirected Hypergraphs

A walk in a hypergraph is the movement through a hypergraph, alternating between nodes and adjacent edges in sequence $(v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n)$. The same nodes and edges cannot appear consecutively. And the movement between nodes can only be to other nodes which share edge(s) with the current node.

Random walks (a particular case of Markov random chain) is used in this application with random steps being taken from one node to another, where each step is completely independent for the last step. The behaviour is determined by a transition probability matrix $\mathcal{P}$, which gives the probability of transitioning to another disease based on a transition matrix e.g. probability of B to follow A.

The transition is in the form of a hyperedge which is connects two specified nodes. Such that given a node u, a hyperedge is chosen, based on the transition matrix, which is incident to u with a probability proportional to $w(e)$.

2.4.3 Successor Detection

Successor detection works similarly to Random Walks on undirected graphs. It is used to find diseases which are likely to follow other diseases, by storing the probabilities of transitioning between disease nodes. This is calculated using the incidence matrix M_D.

In the case of successor detection we only consider the transition from a tail to a head. This is to see which disease is likely to be the next observed.

We can use Eqn 5 to find the probability of there being a transition between two nodes:

$$p(u, v) = \sum_{e \in \mathcal{E}} w(e) \frac{m_-(u, e)}{d_-(u)} \frac{m_+(v, e)}{\delta_+(e)}. \tag{5}$$

Where:

- u is the current node position
- v is the node to be transitioned to
- $m_-(u, e)$ = 1 if node u has a edge e stemming from it (a tail is connected to node u)
- $m_+(v, e)$ = 1 if node v has the head of an edge e connected to it
- $d_-(u)$ = the sum of all possible contributions to u
- $\delta_+(e)$ = the number of nodes connected to the edge e via the edges head (this will always be 1)
- $\frac{m_-(u,e)}{d_-(u)} \frac{m_+(v,e)}{\delta_+(e)}$ = the row normaliser

2.4.4 Successor Detection - PageRank

To calculate the PageRank of the directed hypergraph, the left Eigenvector of the successor transition probability matrix can be calculated. We can perform this the same way as normal Eigenvector centrality as seen in Section 2.4.1 except we must now ensure that we perform the left eigenvector calculation as the transition matrix is not symmetrical as it is in the undirected case. And we must normalise the eigenvector after the calculation so that the columns sum to equal 1. No elements in the transition probability matrix should equal zero as the matrix must be irreducible, but to resolve this we can set 0 elements to a small value such as $1e^{-6}$).

2.4.5 Predecessor Detection

Similar to successor detection, predecessor detection is calculated in the form of a probability transition matrix where the row is the starting node and the column is the node transitioned to. In the case of predecessors we look only at transitions from head to tail. Compared to 5, the operation sign is flipped ($+$ becomes $-$ and vice versa) to give Eqn 6:

$$p(u, v) = \sum_{e \in \mathcal{E}} w(e) \frac{m_+(u, e)}{d_+(u)} \frac{m_-(v, e)}{\delta_-(e)}. \tag{6}$$

3 Datasets

The Secure Anonymised Information Linkage (SAIL) is a database that was created to facilitate the improvement of our understanding of multimorbidity [6].

In this project we used the Wales Multimoribidity e-Cohort (WMC) SAIL DataBank which is comprised of 2,178,938 Welsh patients longitudinal records stratified by Welsh Index of Multiple Deprivation (WIMD) and sex. These records are from between 1st January 2000 and 31st December 2019. Historical electronic health records (EHRs) were included prior to 2000. An individual was recorded to have a disease defined by the Charlson Comorbidity Index (CCI) if it was diagnosed prior to 31st December 2019. Age was defined from 1st January 2000. Individuals were only selected if they were 20 years old or older as of 1st January 2000.

DateTime stamps are included within these records. Records were terminated due to period end (31st December, Welsh residency break or death). It should be noted that death was only recorded if it occurred within the period of analysis, that is prior to December 31st 2019, this can lead to right censoring issues.

Datasets were curated into the CCI, which is formed from ICD-10 codes from hospital admission data grouped into a maximum of 16 conditions. We used 13 groupings of the 16 conditions to prevent condition crossover.

4 Methodology

4.1 Mortality

Mortality (MORT) nodes could be integrated into the hypergraph in various ways. In this section we outline 6 potential routes for adding mortality into the hypergraph as end state nodes.

Initially we gave the patients with one singular disease self-looping hyperarcs, so that these patients are represented within the population.

Table 1 outlines 6 potential ways of including mortality into the directed hypergraph, using two example individuals I and J. Individuals I has an ordered disease set of $\{A, B, C, D\}$ while individual J has ordered disease set $\{A, B\}$. We also further assume that individual I died before the end of the period of analyses while individual J was still alive. Note that n represents the number of total diseases in the hypergraph.

Table 1: 6 different ways to include mortality as nodes into the directed hypergraph.

Mort Type	Description	Extra Hyperarc		Additional Hyperarcs
		Individual I	Individual J	
0	1 MORT node M 0 ALIVE nodes	$A \wedge B \wedge C \wedge D \to M$	No extra hyperarc as J is still at risk of continuing progression.	$\sum_{d=1}^{n}\binom{n}{d}$
1	1 MORT node M 1 ALIVE node S	$A \wedge B \wedge C \wedge D \to M$	$A \wedge B \to S$	$\sum_{d=1}^{n}2\binom{n}{d}$
2	n MORT nodes $M_i, i = A, B, \ldots$ 0 ALIVE nodes	$A \wedge B \wedge C \wedge D \to M_D$	No extra hyperarc as J is still at risk of continuing progression.	$\sum_{d=1}^{n}d\binom{n}{d}$
3	n MORT nodes $M_i, i = A, B, \ldots$ 1 ALIVE node S	$A \wedge B \wedge C \wedge D \to M_D$	$A \wedge B \to S$	$\sum_{d=1}^{n}(d+1)\binom{n}{d}$
4	n MORT nodes $M_j, j = 1, \ldots, n$ 0 ALIVE nodes	$A \wedge B \wedge C \wedge D \to M_4$	No extra hyperarc as J is still at risk of continuing progression.	$\sum_{d=1}^{n}d\binom{n}{d}$
5	n MORT nodes $M_j, j = 1, \ldots, n$ 1 ALIVE node S	$A \wedge B \wedge C \wedge D \to M_4$	$A \wedge B \to S$	$\sum_{d=1}^{n}(d+1)\binom{n}{d}$

We generate 10 fictitious patients for analysing hypergraphs with mortality included, alongside their progressions we provide whether they died within the period of analyses (as given in Table 2).

Table 2: Fictitious patient data disease progressions with hypothetical diseases A, B and C and whether mortality occurred within the study period.

Patient Number	Initial Disease	Secondary Disease	Full Disease Set	Mortality (Died = 1)
1	A	A, B	A, B, C	1
2	A	A, B	A, B, C	1
3	A	A, B	A, B, C	1
4	C	C, A	C, A, B	0
5	B	B, C	-	1
6	B	-	-	0
7	C	-	-	0
8	B	B, A	A, B, C	0
9	B	B, A	-	0
10	A	A, C	-	1

4.1.1 Mortality Comparison Methods

We calculate both predecessor and successor PageRank normalised scores and hyperedge Eigenvector centrality for all the mortality types for the fictitious population given in Table 2 alongside without mortality included as an end state. These scores can be used to quantitatively compare how mortality type effects these scoring systems.

4.1.2 Mortality End State Node Corrections

Self-connections are heavily weighted so we could consider distributing the weights based on degree *(so hyperarc weight vs degree).*

We changed the diseases with self-loops (where a person only has one disease in their entire disease pathway) to either $disease \rightarrow ALIVE$ or $disease \rightarrow MORT$ depending on whether mortality occurred in the period of analysis. That is disease nodes no longer have self-loops. This had no affect on reducing axis pinning, but led to change in the PageRank scores, such that the diseases have larger differences in PageRanks.

When calculating the edge weights using the modified Sørensen's Dice Coefficient (Equation 2) all the predecessors and successors are included to show similarity in the form of the power set and super set. When including $ALIVE$ nodes it may not be beneficial to have the super set otherwise you could have trajectories such as $\{A, B, ALIVE \rightarrow MORT\}$.

For hyperedge calculation of $\{A, B, MORT\}$ we exclude some of the power and super set as they do not clinically make sense, for example $C(A \rightarrow MORT \rightarrow B)$ cannot happen as you cannot have a condition, die and then have another condition. Similarly $A, B, C, MORT$ from the super set cannot be possible.

Dealing with Dead Ends

A dead end (or a terminal node) is a node with no out-links/edges. The presence of dead ends will cause the PageRank of some or all the nodes to converge to 0 in the iterative computation, including nodes that are not dead ends (disease nodes).

There are 4 methods commonly used to deal with dead ends when calculating PageRank these are named and described below along with visual representations in Figure 2.

To try to resolve this we implemented three correction methods to deal with end state nodes these were loop, loop-all and remove. We also observed the effect on PageRank when the power set is included without the super set in the denominator of the Sørensen Dice Coefficient calculation. Finally we tested the mortality model on the SAIL CCI patients to observe the effect mortality has on disease centrality.

Remove Correction

To try to correct for the end state nodes (i.e. $MORT$ and $ALIVE$) we first implemented the Remove correction for PageRank. This was done by calculating the successor and predecessor PageRank with and without mortality included. With both of these PageRank types we could then calculate a 'corrected' PageRank score which balances out the effect of having the dead end nodes. We did this by summing the non-mort PageRank successor and the MORT type 1 PageRank successor for each disease and end state node, then we normalise these totals. This can then repeated with the predecessor PageRanks.

Loop Correction

To perform loop correction we simply need to add self-loops to each end state node. This can be done by adjusting the incidence matrix so that the mortality and ALIVE nodes each have a self loop.

Loop All Correction

Similar to the loop correction we implemented the loop all correction, which applies self-loops to *all* of the nodes rather than just the end state nodes.

Complete vs Power Set Sørensen-Dice Coefficient

We compared the complete Sørensen-Dice coefficient to the power set Sørensen-Dice Coefficient to observe if including the ALIVE node meant the super set should not be included. We performed this comparison on all the correction types, alongside the standard mortality 1 and the no mortality PageRank calculation.

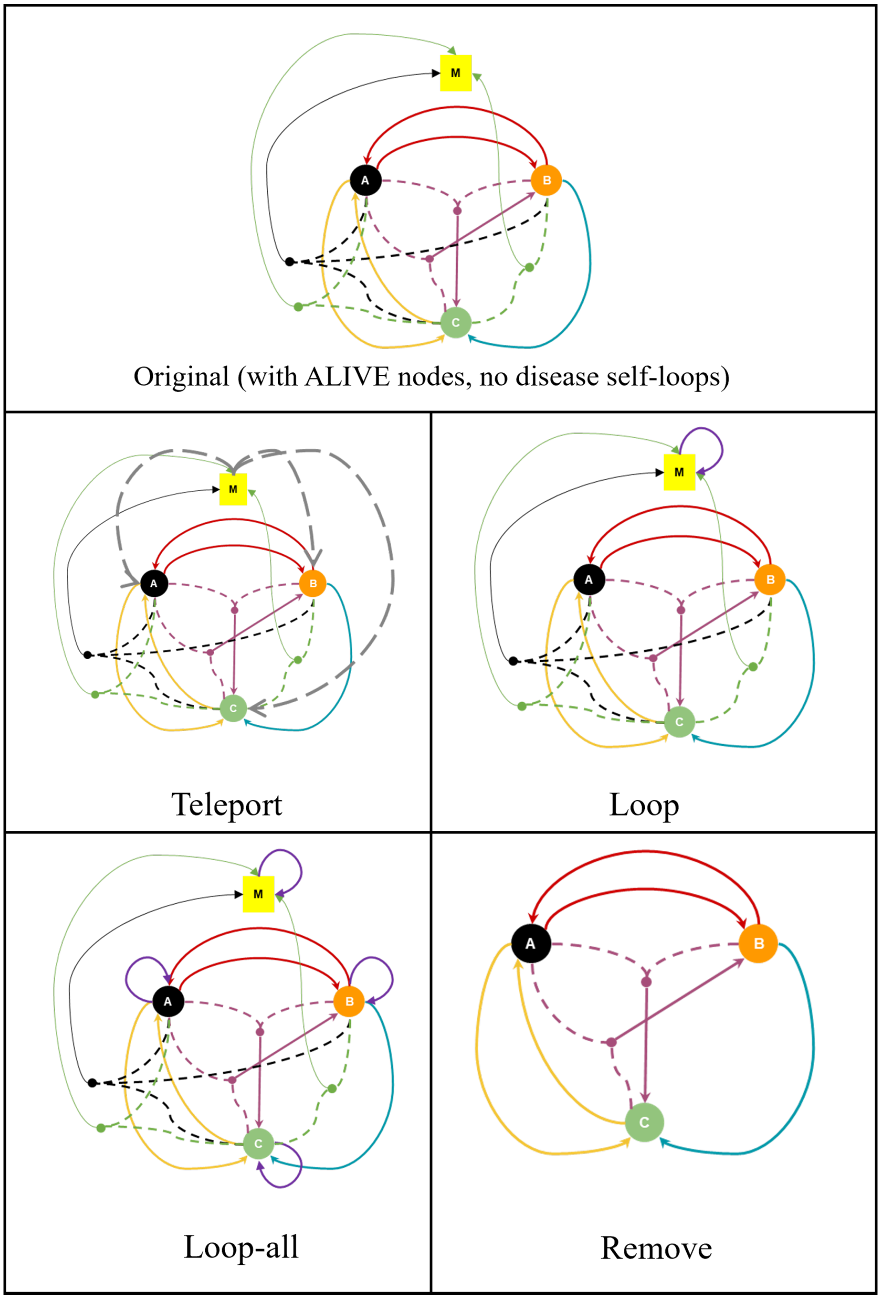

Figure 2: Four methods to deal with dead ends with PageRank: **Teleport:** All dead ends have a hyperarc connection to all the non-dead end nodes. This method confuses real world trajectories as this means hyperarcs such as $MORT \rightarrow DISEASE$ would be present, for this reason we do not implement this method. **Loop:** Self-loops are added to each dead end node. **Loop-all:** All nodes are given self-loop. **Remove:** Dead end nodes are removed iteratively in a graph until no dead ends remain, PageRank is calculated and then dead ends are iteratively reintroduced and PageRank is recalculated until all nodes are restored. *Note:* For visual simplicity we exclude the ALIVE node.

4.2 Temporal Hypergraphs

In this workstream we explored the time between diseases or disease states. Temporality can be useful in healthcare research as it supplies another variable that can assist in tasks such as prediction and pattern-finding. For example, we may generally know that Myocardial Infarction (MI) follows Coronary Pulmonary Disease (CPD) but if we also know roughly how long until MI occurs treatment could be informed to reduce side effects by better intervention timing.

We explored time by looking at the trends of age when different diseases occur, observing the time between different disease states (number of diseases as a state), and calculating the average time the population spend on each hyperarc/disease state. We also discuss a few methods of how we could go about implementing temporality into hypergraphs.

4.2.1 In and Out Graphs

To look into the time between different disease states we plotted a variety of graphs, which we denote as 'In and Out Graphs'. These graphs are very similar to survival curves showing the percentage of people in a disease states over time rather than survival over time.

4.2.2 Hyperarc Delta

We could use time within our hypergraphs as an additional variable to find pathway patterns. Though we must consider that by making the hyperarc weights be equal to the time between disease states (hyperarc delta), we lose the prevalence weighting calculated in Eqn 4. Therefore, it might be worthwhile further considering if prevalence and time could be merged together effectively as one weighting, or whether we could utilise some form of edge weight embedding.

We explored the hyperarc deltas across the population. This work was undertaken in SAIL and as such currently remains in the trusted research environment (TRE).

The shortest path from source to target node with temporality, provides us with another distance metric that consider the length of time rather than just the prevalence or importance of nodes. We considered different ways we could implement temporality into our directed hypergraph and describe below a few methods we could use to calculate the shortest path between two nodes using the time between diseases as the hyperarc weight. Where we define the *source node* as the start node and *target node* as the end node. The time between (hyperarc delta) is the time between the final tail node to the head. There are a few different ways we can look at the shortest path, we can look at the shortest path at each degree or we can look at the overall shortest path.

We add time to the fictitious examples in Table 3. Note that the ALIVE nodes are labelled with node S and this is the time from the last disease to the end of the study period. Another way we could consider the ALIVE node is the time from the last disease to the alive final state being set as 0, this deals with the variation caused the study period being set. You could argue that the ALIVE node might not provide useful information in this temporal setting. Additionally, the $ALIVE$ nodes temporality is determined by the right censor.

Table 4 shows the hyperarc delta for each possible hyperarc (with directionality in the tail nodes) using the data from Table 3. Figure 3 shows how this hypergraph would appear, we exclude the ALIVE and MORT nodes to improve visualisation readability.

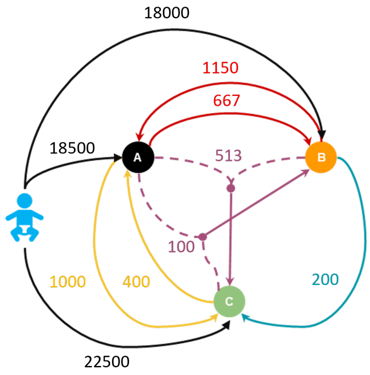

Figure 3: Directed hypergraph with average deltas on the hyperarcs. MORT nodes excluded for simplicity.

Table 3: Fictitious patient data disease progressions with hypothetical diseases A, B and C and whether mortality occurred within the study period (Mortality = M, Alive = S). Row beneath each patient provides fictitious time (days) between the last disease to the left of the arrow to the right of the arrow.

Patient Number	Deg 2	Deg 3	Deg 4	Deg 5
1	Birth → A 21000	Birth, A → B 1000	Birth, A, B → C 200	Birth, A, B, C → M 300
2	Birth → A 15000	Birth, A → B 500	Birth, A, B → C 1000	Birth, A, B, C → M 1500
3	Birth → A 11000	Birth, A → B 500	Birth, A, B → C 700	Birth, A, B, C → M 100
4	Birth → C 19000	Birth, C → A 400	Birth, C, A → B 100	Birth, C, A, B → S 50
5	Birth → B 31000	Birth, B → C 200	Birth, B, C → M 30	-
6	Birth → B 9000	Birth, B → S 5000	-	-
7	Birth → C 26000	Birth, C → S 10	-	-
8	Birth → B 14000	Birth, B → A 300	Birth, A, B → C 150	Birth, A, B, C → S 700
9	Birth → B 18000	Birth, B → A 2000	Birth, B, A → S 5000	-
10	Birth → A 27000	Birth, A → C 1000	Birth, A, C → M 100	-

Table 4: Hyperarcs and their average delta using the 10 fictitious patients (rounded to whole numbers).

Hyperarc	Average Delta
Birth → A	18500
Birth → B	18000
Birth → C	22500
Birth, A → B	667
Birth, A → C	1000
Birth, B → A	1150
Birth, B → C	200
Birth, B → S	5000
Birth, C → A	400
Birth, C → S	10
Birth, A, B → C	513
Birth, A, C → M	100
Birth, B, A → S	5000
Birth, B, C → M	30
Birth, C, A → B	100
Birth, A, B, C → M	633
Birth, A, B, C → S	700
Birth, C, A, B → S	50

Shortest Path Ideas

In the following examples we calculated the shortest path between disease A to Mortality M.

Idea 1

Standard graphs could be used instead of hypergraphs and all pathways from A to M could be summed. However, this does not provide any predecessor conditions that exist before A and this gets the average time between node pairs rather than based on multimorbidity sets. Figure 4 shows an example of how this standard graph might look with three diseases, a birth and a MORT node. We lose a significant amount of information here.

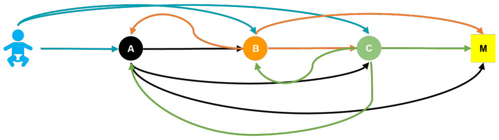

Figure 4: Standard graph representation showing node connections.

You could extend these standard graphs by providing attributes to the edges such that we know how many diseases are present at a specific stage, but this then becomes very similar to hypergraphs - effectively instead of knowing which diseases are tail nodes we would know the number of diseases in the tail.

Idea 2

The final idea looks at graphs in a different way. We retain the order of the diseases, rather than the tail of the hyperarc effectively becoming undirected we keep the directionality. Though strictly no longer hyperarcs, we could think of each patient trajectory to be a set of subgraphs, where only pairwise connections are made. We can store this information as a sort of 3D adjacency matrix where the axis x is the start node, the axis y is the next node and axis z is the degree of the subgraph with the average delta time stored in the depicted position. We then need a 3D matrix for each of the 'hyperarcs' which stores the order the disease trajectory (with 1's where a connection exists and 0's otherwise), such that we can perform a element-wise matrix multiplication to obtain the temporal pathways and navigate the 3D adjacency matrix.

For example, using the 10 fictitious patients again we could have the 3D adjacency matrix below where each 2D matrix (Degree 2 - Degree 5) would be constructed together to form the 3D matrix. *Let E instantiate birth for this example.*

$$\begin{array}{ccccc} & E & A & B & C \\ A & 21000 & 0 & 0 & 0 \\ B & 18000 & 0 & 0 & 0 \\ C & 22500 & 0 & 0 & 0 \\ M & 0 & 0 & 0 & 0 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 1150 & 400 \\ 0 & 667 & 0 & 0 \\ 0 & 1000 & 200 & 0 \\ 0 & 0 & 0 & 0 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 513 & 0 \\ 0 & 0 & 0 & 65 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 633 \end{array}$$

Then if we want to show the pathway of the trajectory $Birth \rightarrow A \rightarrow B \rightarrow C \rightarrow M$:

$$Path = \begin{array}{ccccc} & E & A & B & C \\ A & 1 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 \\ M & 0 & 0 & 0 & 0 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}, \quad \begin{array}{cccc} E & A & B & C \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}$$

For the pathways $Birth, A, C \rightarrow M$ and $Birth, B, C \rightarrow M$, because the penultimate nodes are both C and the last node is M in both, we then take the average of these and supplied them in our 3D adjacency matrix. This method also allows us to find the shortest path regardless of where our source node first occurs. As we can ignore the prior 2D matrices of a patient's pathway which do not have the source node in the x axis.

We can then calculate the shortest path from A to M at each degree, alongside the shortest overall path. At degrees 2 and 3 no paths between A and M exist.

Degree 4: $Birth, A, C \rightarrow M$ is the only node degree four hyperarc which has A in the tail and M in the head so we can use the matrices to find the routes: ($Birth, A, \rightarrow C$ = 1,000 days) + ($Birth, A, C \rightarrow M$ = 100 days) = 1,100 days.

Degree 5: $Birth, A, B, C \rightarrow M$ is the only degree five hyperarc which has A in the tail and M in the head. The shortest path at degree 5 is then found from the hyperarcs: ($Birth, A \rightarrow B$ = 667 days) + ($Birth, A, B \rightarrow C$ = 513 days) + ($Birth, A, B, C \rightarrow M$ = 633 days) = 1,813 days.

Overall shortest path: $A \rightarrow C \rightarrow M$ = 1,100 days.

The clinical utility of this could be to provide a form of knowledge base to show average time between diseases, multimorbidities and mortality outcomes. Clinicians could use this to inform decision making practices for treatment and palliative care, with the aim to improve quality of life and health outcomes.

5 Results and Discussion

5.1 Streamlit Multimorbidity Hypergraphs Applet

We wanted to create an interactive applet that explains and demonstrates the use of hypergraphs for multimorbidity analysis at a high-level. We achieved this and reproduced previous work [3], as described in the Background (section 2) of this report. This applet aims to be a tool that explains hypergraphs in a multimorbidity setting, how to calculate Eigenvector centrality, probability transition matrices, and PageRank using hypergraphs.

The code generates a fictitious set of patients where the user can select the number of fictitious diseases (alphabetically labelled) and the number of patients. The applet displays the maximum number of possible edges for various different graphs (standard undirected, standard directed, undirected hypergraph, directed B-hypergraph, and directed BF-hypergraph), informing the user of the complexity and computational growth caused by increasing the number of diseases/nodes in graphs. The applet has various pages. The first page gives an explanation of multimorbidity, graphs and hypergraphs, setting the scene. The second page shows a HypernetX visualisation of the generated population's disease trajectories as a undirected hypergraph and then takes the reader through a step by step explanation of how to calculate hyperedge weights and Eigenvector centrality. The third page gives a B-hypergraph NetworkX visualisation of the population's trajectories, then explains how to calculate hyperarc weights, successor and predecessor transition matrices and PageRank. The final page allows the user to input an alphabetical disease or disease set, which then prints out the most likely disease successors given the fictitious population's trajectories. The user can adjust the number of possible trajectories and the maximum degree (diseases) to show.

There are very few resources available explaining hypergraphs, and even fewer that demonstrate hypergraphs in a health setting. We hope that this applet will be useful to demystify hypergraphs and explain how they could be useful in the multimorbidity space.

To use the applet visit the `Hypergraphical` GitHub repository here and follow the instructions in the README.md to run it locally.

5.2 Mortality Results

5.2.1 PageRank

Mortality Types Predecessor PageRank

We give the predecessor PageRank scores in Table 5. When mortality is not included node B has the highest PageRank score out of all of the diseases, suggesting this node is the most likely to be a predecessor and node C has the lowest score so is the least likely. When we look across the mortality types in Table 5, we can see that none of the mortality types follow this trend. Instead, we see that out of the three disease nodes, node C is given the highest PageRank score in 5 out of 6 MORT types. When mortality is included node B often obtains the lowest PageRank score instead of the highest (5/6 times). All MORT types except MORT 3 follow the same pattern with node C having the highest predecessor score, node A the second highest, and node B the lowest.

Looking at the predecessor PageRank score for the different MORT nodes, some of the MORT nodes have a score more than 0 which should not be possible as mortality is always the final state.

Mortality Types Successor PageRank

Next we can look at how mortality has affected the successor PageRank score (Table 6). Here node C has the highest PageRank score, followed by node B and then A.

When mortality is included the successor PageRank score for all of the disease nodes becomes 0, providing no information of the probability of a disease being a successor. Mort type 0 suggests that mortality is the only end state/successor node, however this is not true as some pathways end with a disease or disease set. Mort type 1 gives

Table 5: Predecessor PageRank for the 6 different mortality type.

Node	Predecessor PageRank						
	No Mort	$Mort_0$	$Mort_1$	$Mort_2$	$Mort_3$	$Mort_4$	$Mort_5$
B	0.403	0.213	0.215	0.16	0.331	0.191	0.174
A	0.363	0.26	0.329	0.196	0.314	0.234	0.267
C	0.234	0.527	0.456	0.398	0.323	0.474	0.37
$MORT$	-	0	0	-	-	-	-
$ALIVE$	-	-	0	-	0	-	0
$MORT_A$	-	-	-	0.141	0.012	-	-
$MORT_B$	-	-	-	0.105	0.021	-	-
$MORT_C$	-	-	-	0	0	-	-
$MORT_1$	-	-	-	-	-	0.101	0.189
$MORT_3$	-	-	-	-	-	0	0
$MORT_2$	-	-	-	-	-	0	0

a greater picture of the connectivity of the mortality and the ALIVE node, with the $MORT$ node having around 1/3 more connectivity than the $ALIVE$ node.

The hierarchy of MORT node PageRank in MORT type 2 suggests that $MORT_C$ has the highest connectivity, followed by $MORT_B$ and then $MORT_A$. However, when an $ALIVE$ node is introduced in MORT type 3 this hierarchy shifts with the PageRank for $MORT_B$ and $MORT_C$ becoming very similar. This is in a case where there is no connectivity to $MORT_A$, so you might imagine that if there was connectivity to $MORT_A$ the PageRank score for the three MORT nodes could become insignificantly different.

Mort type 4 and 5 have similar issues to that of 2 and 3, where adding an $ALIVE$ node brings the MORT node scores closer together.

Table 6: Successor PageRank for the 6 different mortality type.

Node	Successor PageRank						
	No Mort	$Mort_0$	$Mort_1$	$Mort_2$	$Mort_3$	$Mort_4$	$Mort_5$
B	0.249	0	0	0	0	0	0
A	0.241	0	0	0	0	0	0
C	0.51	0	0	0	0	0	0
$MORT$	-	1	0.605	-	-	-	-
$ALIVE$	-	-	0.395	-	0.336	-	0.191
$MORT_A$	-	-	-	0.085	0.162	-	-
$MORT_B$	-	-	-	0.272	0.252	-	-
$MORT_C$	-	-	-	0.642	0.25	-	-
$MORT_1$	-	-	-	-	-	0.038	0.119
$MORT_3$	-	-	-	-	-	0.419	0.338
$MORT_2$	-	-	-	-	-	0.542	0.352

Overall, you could argue that when mortality is included the successor PageRank score does not clinically make sense and adds little value, it suggests a rough proportion of people with an end state of alive or mortality depending on the number of conditions they have or their final disease before mortality. But as $MORT_A$, $MORT_B$ and $MORT_1$ have no connectivity in the example of our ten fictitious patients it may be worthwhile considering a PageRank correction since these scores should not be more than 0.

Additionally, PageRank is used to show node importance based on connectivity, but if we want to learn what pathways are likely to be succeeded with specific outcomes connectivity may not be the best way to determine this.

Hyperedge Eigenvector Centrality

Table 7 gives us the Eigenvector centrality scores for each of the hyperedges present in the dual hypergraph representing the 10 fictitious patients.

Table 7: Hyperarc Eigenvector centralities for the 6 different mortality type.

Hyperarc	Eigenvector Centrality						
	No Mort	$Mort_0$	$Mort_1$	$Mort_2$	$Mort_3$	$Mort_4$	$Mort_5$
$A \rightarrow A$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$B \rightarrow B$	0.175	0.163	0.140	0.195	0.170	0.179	0.160
$C \rightarrow C$	0.274	0.132	0.036	0.071	0.037	0.139	0.042
$A \rightarrow B$	0.376	0.000	0.000	0.000	0.000	0.000	0.000
$C \rightarrow A$	0.030	0.157	0.000	0.172	0.000	0.171	0.000
$B \rightarrow C$	0.343	0.397	0.356	0.395	0.398	0.415	0.393
$B \rightarrow A$	0.113	-	0.195	-	0.211	-	0.212
$A \rightarrow C$	0.463	-	-	-	-	-	-
$A, B \rightarrow C$	0.555	0.332	0.214	0.312	0.212	0.348	0.234
$A, C \rightarrow B$	0.315	0.231	0.130	0.272	0.144	0.253	0.148
$A \rightarrow MORT$	-	0.000	0.000	-	-	-	-
$A, B, C \rightarrow MORT$	-	0.000	0.432	-	-	-	-
$B \rightarrow MORT$	-	0.498	0.451	-	-	-	-
$B, C \rightarrow MORT$	-	0.339	0.349	-	-	-	-
$C \rightarrow MORT$	-	0.498	0.444	-	-	-	-
$A, B, C \rightarrow ALIVE$	-	-	0.206	-	0.237	-	0.235
$A \rightarrow MORT_A$	-	-	-	0.000	0.000	-	-
$A, B, C \rightarrow MORT_C$	-	-	-	0.000	0.425	-	-
$B \rightarrow MORT_B$	-	-	-	0.491	0.433	-	-
$B, C \rightarrow MORT_C$	-	-	-	0.324	0.317	-	-
$C \rightarrow MORT_C$	-	-	-	0.503	0.420	-	-
$A \rightarrow MORT_1$	-	-	-	-	-	0.000	0.000
$A, B, C \rightarrow MORT_3$	-	-	-	-	-	0.000	0.406
$B \rightarrow MORT_1$	-	-	-	-	-	0.477	0.432
$B, C \rightarrow MORT_2$	-	-	-	-	-	0.282	0.283
$C \rightarrow MORT_1$	-	-	-	-	-	0.504	0.458

Interestingly, hyperarcs with A, B, C leading to any form of $MORT$ have a centrality value of 0, unless an ALIVE node is included which then makes centrality equal more than 0.

5.2.2 Mortality Type Selection

Mortality changes the ordering of predecessor disease PageRanks, flipping it so that disease C is more of a predecessor than the other two nodes.

Including $ALIVE$ nodes brings the predecessor PageRanks closer to each other and the $MORT$ and $ALIVE$ nodes closer to each other. If you include ALIVE nodes, then they will always be present in the superset even though an ALIVE node might be replaced with a MORT node. Effectively, this means that ALIVE nodes will end trajectories but clinically this is not true. That means if we are to include $ALIVE$ nodes we should use the Sørensen-Dice Coefficient which only includes the power set.

Mort type 1 provided us with symmetry on the terminal/end nodes, with a single $MORT$ node and one $ALIVE$ node. The $ALIVE$ node balances the effect of the MORT node(s) allowing trajectories to have an end state of alive as well as mortality. Mort type 1 also uses minimal computational resources compared to the other types 2-5.

The two types (MORT type 1, and MORT type 3) are likely the best routes to focus on as they make the most sense clinically and mathematically. It may also be seen as common knowledge that the more multimorbidities you have, the more likely mortality is going to happen. Therefore, having multiple MORT nodes with the sum of diseases at mortality is probably computationally not worthwhile.

We decided to focus on MORT type 1 to observe the effects on the correction.

5.2.3 PageRank Correction Comparison

In Table 8 the PageRanks for the different correction types are shown for the 10 fictitious patients.

Table 8: PageRank scores for Mortality type 1 with 3 different corrections applied.

Predecessor PageRank			
Disease	LOOP ALL	LOOP	REMOVE
A	0.503	0.428	0.400749064
B	0.359	0.366	0.378277154
C	0.138	0.206	0.220973783
MORT	0	0	0
ALIVE	0	0	0
Successor PageRank			
Disease	LOOP ALL	LOOP	REMOVE
A	0	0	0.119
B	0	0	0.1225
C	0	0	0.2585
MORT	0.575	0.524	0.2095
ALIVE	0.425	0.476	0.2905

Table 9 shows the difference in the PageRank scores when the complete Sørensen-Dice coefficient is calculated vs the Sørensen-Dice coefficient without the super set in the denominator. Here we can see that the removing the super set from the denominator has a minimal impact on the PageRank scores.

Table 9: The absolute difference in PageRank when using the complete vs the power set Sørensen-Dice Coefficient.

Predecessor PageRank					
Disease	Loop All	Loop	Remove	Standard Mort 1	Standard No Mort
A	0.008	0.001	0.002	0.001	0.007
B	0.002	0.015	0.009	0.015	0.013
C	0.006	0.014	0.007	0.014	0.006
MORT	0	0	0	0	0
ALIVE	0	0	0	0	0
Successor PageRank					
Disease	Loop All	Loop	Remove	Standard Mort 1	Standard No Mort
A	0	0	0.019	0	0.038
B	0	0	0.0015	0	0.003
C	0	0	0.0205	0	0.041
MORT	0.186	0.051	0.0275	0.055	0
ALIVE	0.186	0.051	0.0275	0.055	0

Given that the Remove correction gave the clearest distinction between the disease PageRanks and was best at reducing axis pinning (convergence of the successor PageRank to 0) of the disease nodes, we decided to use this method as our PageRank correction.

Loop Correction

Despite the hyperedge weights changing when the loop correction was applied, the PageRank scores for the disease nodes remained the same and still pinned to the y-axis. The end state nodes had different PageRank scores than without the end state node self-loops, such that their PageRanks gained a larger difference (though remained pinned to the x-axis). This trend remained even when the prevalence was changed for each of the end state nodes, such that the prevalence was either the number of times $ALIVE$ and $MORT$ were the final states and when we assigned the prevalence of these end state self-loops as 1.

To experiment, we set the prevalence of the MORT nodes to be very large (e.g. $1e^8$) which caused the nodes to start clustering towards the transitive line rather than pinning to the x and y axis. Albeit the MORT nodes should remain as successor nodes due to their final states. However, when we introduce the self-loops on the end state node they technically become capable of being predecessors to themselves. From this we can see that adding self-loops on end state nodes could be beneficial to prevent the disease nodes successor PageRanks from converging to 0.

Loop-all Correction

First we tried the prevalence count for each self-loop (hyperarc) to equal one. This minimally changed the PageRank values for all of the nodes, not resolving axis pinning.

Next we made the prevalence count of the disease self-loops to be the number of times each disease occurs (in the progression sets and final state e.g. for A then $A \to C$ then $A, C \to MORT$, the prevalence of A would be 3). The end state node prevalence was counted based on the number of people who end up in each state, the progression steps are not included. Interestingly, this has similar PageRank $MORT$ and $ALIVE$ scores that were found without correction, but the disease nodes still had y-axis pinning with differences in predecessor PageRank score for the diseases widening.

Charlson Hypergraph Outcomes

In this section we present the results of mortality type 1 with the remove correction on PageRank using the 13 Charlson conditions.

First, we compared the top 28 hyperedge (Figures 5 and 6) and hyperarc (Figures 7 and 8) weights with and without mortality type 1.

In terms of hyperedge weightings, we gained a distinction between diseases alone (DISEASE, ALIVE) and diseases that have successors (DISEASE). Looking at the highest hyperedge weighting we can see that CPD is the most prevalent Charlson disease in this dataset. Having the MORT nodes shows that CPD may occur more often alone than with other diseases. However, this weighting could also be attributed to the fact the disease sets containing $ALIVE$ do not have any super sets and so the denominator of the Sørensen-Dice coefficient may be smaller. In fact, the only diseases that have a lower hyperedge weighting than their single disease set alternative are Connective Tissue Disease (CTD), Peptic Ulcer Disease (PUD), and MI.

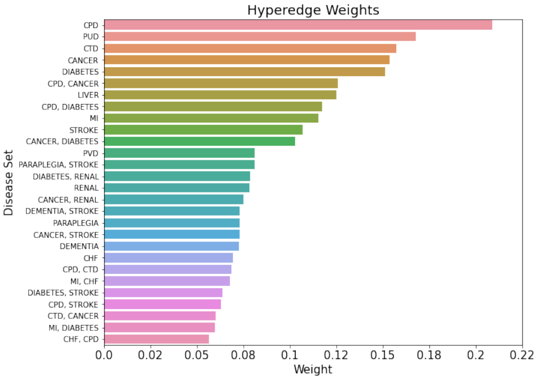

Figure 5: Hyperedge weights for top 28 hyperedges when mortality is not included.

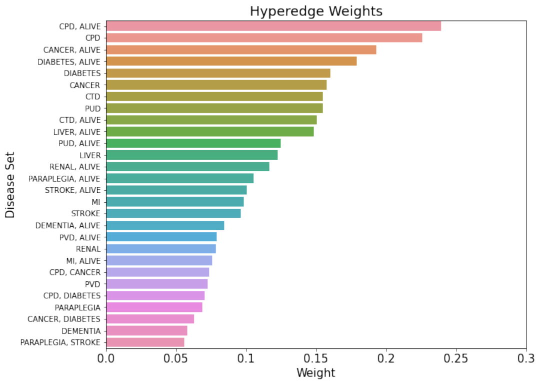

Figure 6: Hyperedge weights for top 28 hyperedges when mortality type 1 is included.

In the top hyperarc weightings (Figures 7 and 8), the self-looping single diseases have been replaced with the $DISEASE \to ALIVE$ counterparts. This has also led to a shift in the weightings, again this may be due to there being no supersets for disease sets containing $ALIVE$.

Table 10 shows the successor and predecessor PageRanks for all of the 13 Charlson conditions (and the mortality outcomes) with and without mortality, alongside the remove corrected PageRanks which are calculated using the PageRanks with and without mortality. Where, for example, the remove corrected predecessor PageRank is calculated for CPD by: $\frac{0.155+0.158}{\sum NoMortPredecessorPageRank + \sum Mort1PredecessorPageRank}$.

Table 10 can be visualised by a scatterplot where the y-axis is the predecessor PageRank and the x-axis is the successor PageRank. Figure 9 shows the disease node PageRanks without mortality included, Figure 10 shows the disease node PageRank with mortality type 1 and the remove correction function. You can see that the underlying topology of the diseases is partially preserved between these two figures.

The remove function we have applied appears to be an effective way to show the PageRank of the diseases when $MORT$ and $ALIVE$ nodes are included. However, the diseases when the MORT nodes are included the disease nodes

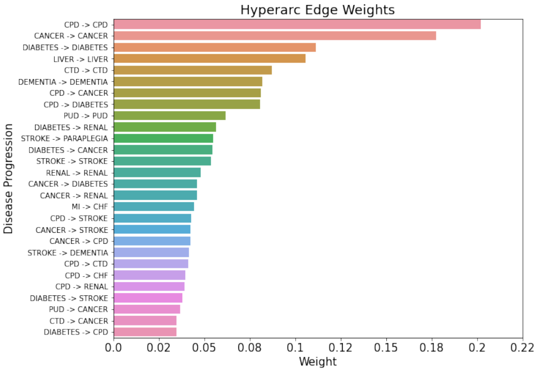

Figure 7: Hyperarc weights for the top 28 hyperarcs when mortality is not included.

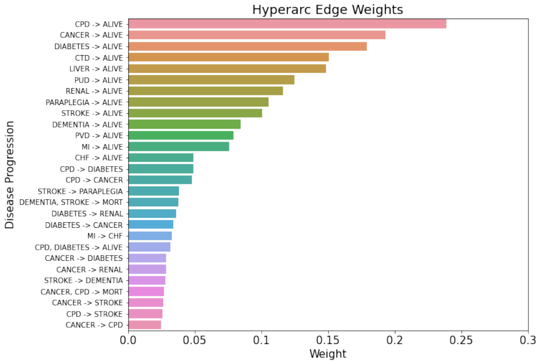

Figure 8: Hyperarc weights for the top 28 hyperarcs when mortality type 1 is included.

Table 10: PageRank with and without mortality included on the 13 Charlson conditions, with the Remove correction applied. Note, we use short terms for the conditions e.g. Congestive Heart Failure (CHF) and Peripheral Vascular Disease (PVD).

Disease	No Mort Predecessor PageRank	No Mort Successor PageRank	Mort 1 Predecessor PageRank	Mort 1 Successor PageRank	Corrected Successor PageRank	Corrected Predecessor PageRank
CPD	0.155	0.087	0.158	0.000	0.044	0.156
DIABETES	0.123	0.106	0.139	0.000	0.053	0.131
MI	0.105	0.073	0.090	0.000	0.037	0.097
CANCER	0.104	0.113	0.116	0.000	0.0565	0.110
PUD	0.088	0.033	0.082	0.000	0.017	0.085
STROKE	0.079	0.115	0.091	0.000	0.057	0.085
RENAL	0.067	0.114	0.079	0.000	0.057	0.073
CTD	0.066	0.039	0.056	0.000	0.020	0.061
CHF	0.059	0.108	0.055	0.000	0.054	0.057
LIVER	0.055	0.016	0.029	0.000	0.008	0.042
DEMENTIA	0.043	0.086	0.045	0.000	0.043	0.044
PVD	0.035	0.066	0.039	0.000	0.033	0.037
PARAPLEGIA	0.023	0.045	0.021	0.000	0.022	0.022
ALIVE	0.000	0.000	0.000	0.572	0.285	0.000
MORT	0.000	0.000	0.000	0.428	0.213	0.000

will remain on the transitive boundary or the predecessor portion of this scatterplot. Regardless, we can see that CPD is a predecessor disease likely to appear before other Charlson conditions, whereas CHF is more likely to be a successor.

5.3 Temporality Results - IN and OUT Graphs

First we plotted time from birth to the first recording of any Charlson disease in Figure 11. We trimmed those below 20 and those over 100 years old to prevent small numbers of individuals being shown in the tails of the curve. Here we can see that most people get their first Charlson disease around age 65. Also we see a clear left/negative skew of the curve, which could be due to less people living past 80.

Figure 12 shows 13 curves, each for a different Charlson disease. We show the percentage of the population and the time they have their first recording of each Charlson disease. This is useful to observe which diseases are more prevalent and at what age people first get diagnosed with each Charlson disease. Again we crop the x-axis at 24 and 95 to prevent inclusion of small numbers of patients in our plots. Some things that we can observe from this plot include: CPD occurs at a younger age and more often than other Charlson diseases. Liver disease has the smallest population prevalence with only a slight increase in chance of liver disease at age 60 over ones lifetime. As expected dementia is highly right skewed as it tends to mostly occur in those over 50.

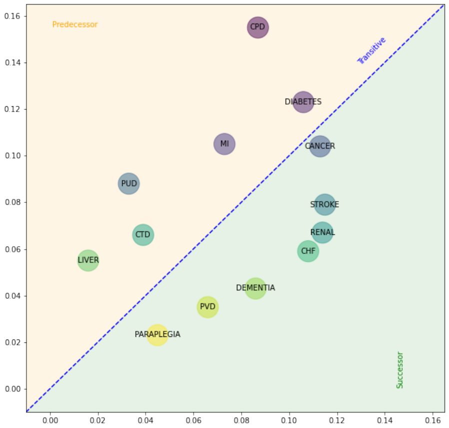

Figure 9: Charlson disease node PageRanks without mortality included.

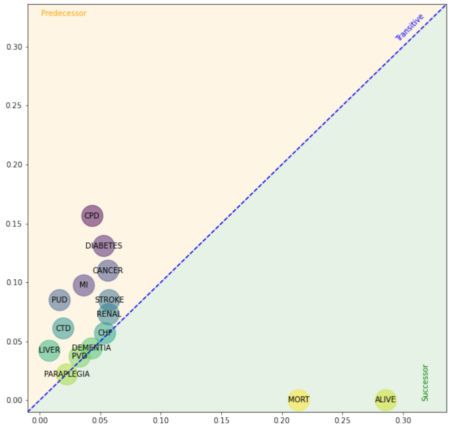

Figure 10: Charlson disease node PageRanks with mortality type 1 included and remove PageRank correction applied.

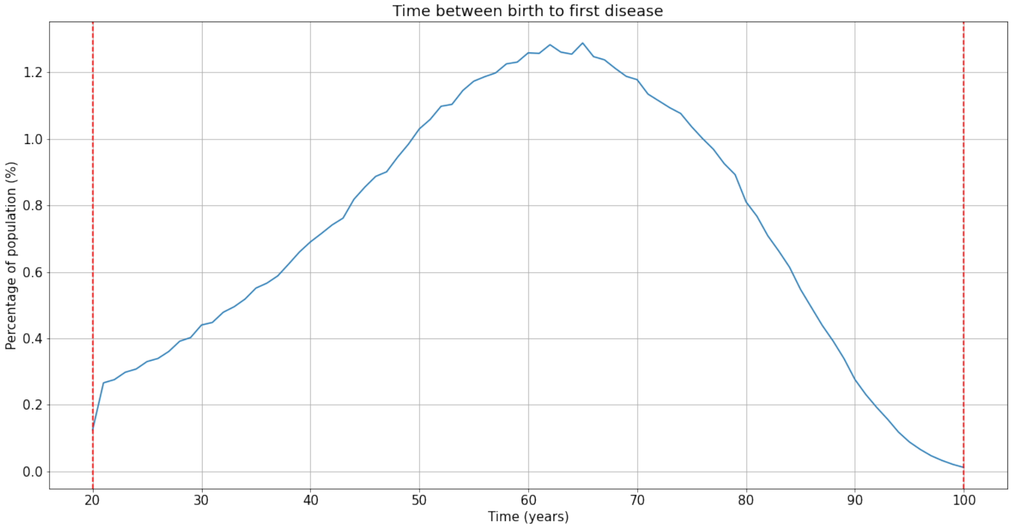

Figure 11: Percentage of population who have their first Charlson disease at specific ages.

Interestingly, in Figure 12 we can see a 'blip' in the PVD curve at around 65 on the x-axis, with what appears to be a surge in people. This could be due to checks being recommended leading to a surge of diagnoses that may have been symptom free or which had minor ignored symptoms, or a data quality issue.

Next, we looked at patterns in temporality based on state changes. Here we define a state change as a change in the number of diseases (disease degree). We coined these 'In and Out graphs', such that we observe trends going in and out of disease states. In particular we look at 4 state changes: one degree in, one degree out, two degree out and two degree in. Of course, we could extend this to cover more than two degrees.

Figure 13 shows the One Degree In graph, where we observe the number of people who come out of the state of having no Charlson diseases (degree zero) to having one Charlson disease (degree one). The red dashed line shows where we

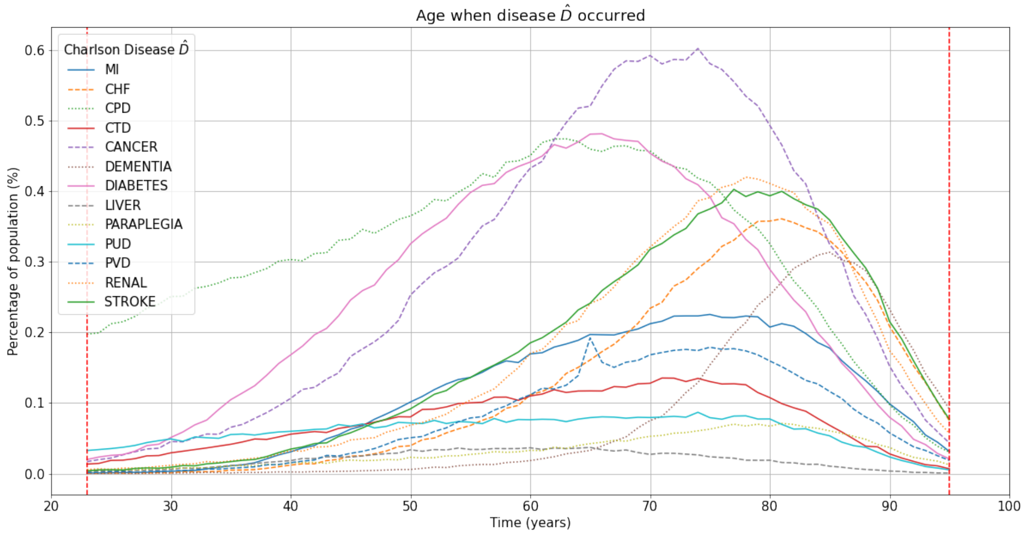

Figure 12: Percentage of population who have their first occurrence of a specific Charlson disease at an age.

trimmed off the graph to protect the small number of individuals in the over 100 year old bracket. The curve begins at ∼58% as the rest of the SAIL cohort (∼42%) does not have a Charlson disease.

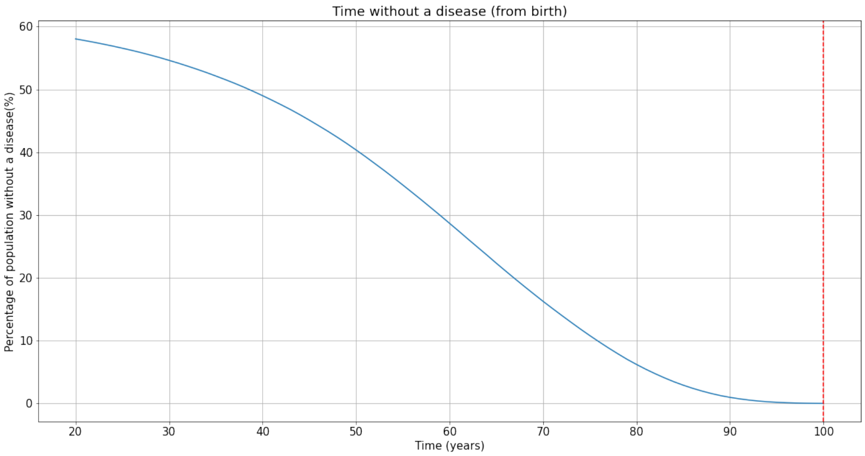

Figure 13: 'One Degree In' - Percentage of population without a Charlson disease over time/age.

In Figure 14 the percentage of the population going from having a specific Charlson disease (each disease is coloured and coded as per the legend) to having that disease plus any other disease. For example, from the CPD curve (green dotted curve) we can see that there is a significant proportion of the population who have CPD as there first disease and there is quite a steep curve initially, where the population will have a second disease 0-30 years after having CPD recorded (where CPD is the first recorded Charlson disease). Whereas there's a relatively small population with only liver disease. CPD, diabetes and cancer seem to be the Charlson diseases that occur most often first in a patient's pathway.

For the 'Two Degree In' (Figure 15) we look at the proportion of the population going from having one Charlson disease to having a second specific disease (shown as $\hat{D}$ in the legend). It is observed that most people with a non-CPD

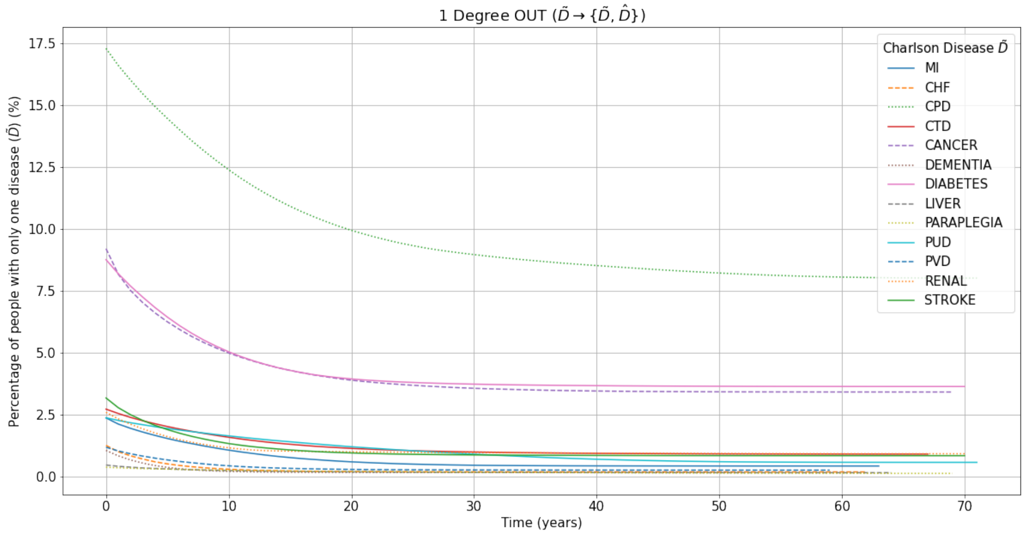

Figure 14: 'One Degree Out' - Percentage of population going from having one Charlson disease to more than one Charlson disease.

disease who gain CPD as their second disease will mostly likely gain it between 0-20 years after their first disease, with the curve continuing until around 50 years and then plateauing.

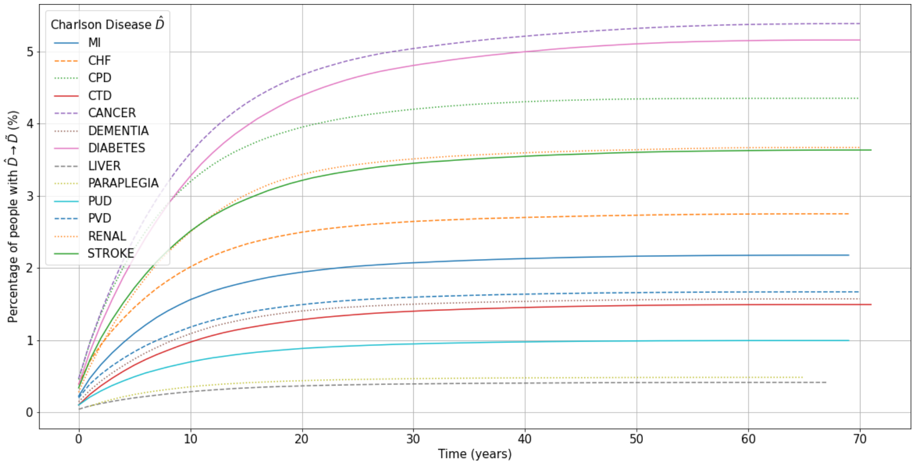

Figure 15: 'Two Degree In' - Percentage of population going from having any Charlson disease except disease $\hat{D}$, to having that disease + disease $\hat{D}$.

Similarly, we first show the 'Two Degree Out' graph with all 13 Charlson conditions (Figure 16). Where we look at moving from degree two to degree three, with a specified disease in the tail component. Again we use the red dashed line to trim the graph to cut off tails where only small numbers of patients would be shown.

With the Two Degree Out graphs we can see the continuing trend that CPD, diabetes and cancer are the most prevalent diseases that come first or second in a patient pathway.

From these In and Out graphs we can see that the different Charlson diseases and the different degree states have quite distinct and smooth curves, suggesting that temporality could be useful to assist with finding patterns within the data.

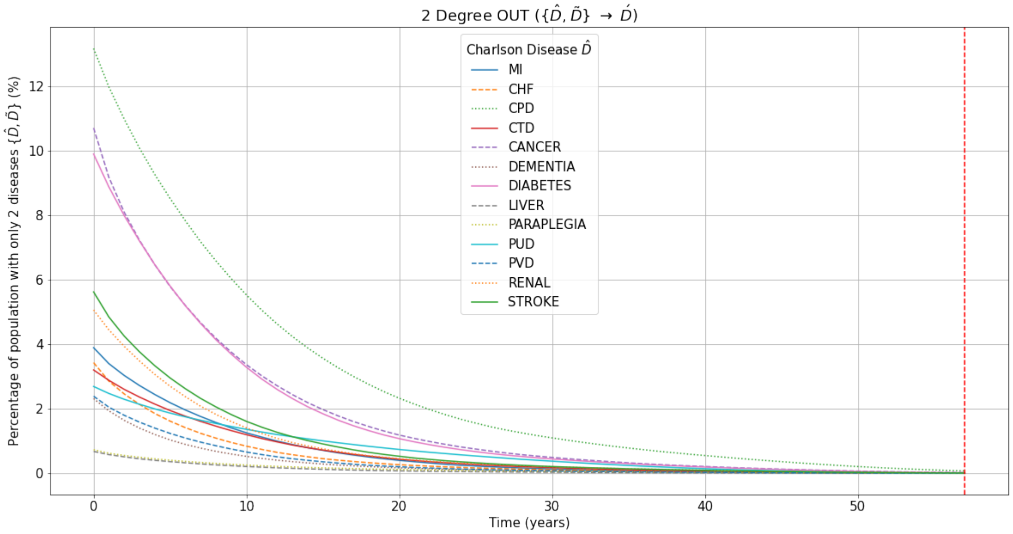

Figure 16: ‘Two Degree Out’ - Percentage of population going from having two Charlson diseases to having those two diseases plus a third disease.

6 Conclusion

In this report, we described how hypergraph methods are continuing to be evolved to use in healthcare settings. We described how mortality can be incorporated as end state nodes to include health outcomes within hypergraphs alongside diseases. We illustrated how mortality affects node and edge weights, alongside providing a hierarchy of disease importance, when end states are included in the form of PageRank successor and predecessor scores. Correction methods to deal with terminal nodes are described and compared, along with a brief comparison of the complete and power set Sorensen-Dice coefficient. Additionally, we assessed disease state temporal patterns and determine that temporality is a worthwhile avenue to explore further. Including temporality in hypergraphs as an additional variable should provide more detailed patterns within the data, such as average time to next disease state. Finally, we produced an interactive Streamlit applet that provides a walkthrough on how to build undirected and directed hypergraphs, alongside their respective centrality scores and a tool to show which disease progressions are likely based on a current disease set and the populations trajectories.

7 Future work

Demographics: The SAIL WIMD contains mortality and demographic data, both rich sources of information which could be implemented into the hypergraph models as variables. Including extra information alongside the first recording of a disease provides a means to stratify the population and create smaller groups. If patients are clustered into groups, the smaller the groups are the potentially better results we can get for tasks such as health outcome prediction, which in turn can enable better treatment/preventative plans to be put in place for individuals rather than a ‘one size fits all’ approach. Similarly to how we implemented mortality into hypergraphs we could include demographics as a variable in hypergraphs.

For example, these could be the starter nodes in the hypergraph such as birth, WIMD, sex enabling more discrete clustering’s of patients. Though it should be noted that we would then have the opposite problem of the dead-end nodes, such that these demographic nodes would have no predecessors. Also, worth considering is how these nodes would be constructed. They would most likely need to be discretised at the level of a population hypergraph, for example binning age groups such that one age node could be those aged 20-30 years.

With both demographics and mortality included within these hypergraphs it might be beneficial to look at the benefits of having these two types of variables. How much information is truly required to make a model useful for clinical utility, how many variables are required within a model to have significantly better results?

Temporality: We calculated the mean time on each hyperarc for the Charlson population. These averages could be implemented into hypergraphs as node or edge attributes or weights.

It's also important to consider if the metrics we use lead to unusual patient trajectory timelines being ignored or deemed as outliers. These distribution tail patients are likely to have patterns that may be important for rare cases and perhaps even in the most severe cases.

Use temporality as edge weights and then calculate Eigenvector centrality to see how the scores differ to the Sørensen-Dice coefficient.

GNNs: Using ML/AI models to predict patient-level outcomes rather than at a population-level. Demographics could be included in the GNN model perhaps by inputting them through a simple multi-layered perceptron rather than as a node in the graph. Node-level (next disease prediction), edge-level (time until next disease prediction) or graph-level prediction (health outcome prediction)?

Other Centrality Measures: In this project we have looked at Eigenvector Centrality to calculate undirected hypergraph node centrality and left Eigenvector Centrality to calculate directed hypergraph node centrality. Some of the other measures that may be worth exploring include: Katz Centrality, Harmonic Centrality, Closeness, K-Nearest neighbours/nodes, Shortest/longest path, Betweeness.

Episodic Events: It would be interesting to perform some exploratory work on looking at finding which conditions/events that are either acute or chronic, this would be useful to determine the effect of these lengths on the overall patient progression. How many times does each disease get recorded per person? What is the average number of times each condition was recorded over the whole study period broken up into years/decades. Does this help us determine whether diseases are episodic or not. Could we use the fact that we know some conditions are acute to decay condition importance over time?

Exploring Other Diseases Groups/Indexes as Nodes: In this report we focused on the CCI, previous work also looked the Elixhauser Index and there is also ongoing work with Caliber. Another potential index is the electronic frailty index to see how multimorbidities stack over time. Interventions and treatments may also be an option for node allocation to see what treatment plans lead to patient outcomes.

Acronyms

CCI Charlson Comorbidity Index. 5–7, 22

CHF Congestive Heart Failure. 17

CPD Coronary Pulmonary Disease. 9, 16, 17, 19, 20

CTD Connective Tissue Disease. 16, 17

EHR electronic health record. 5

GP General Practitioner. 1

MI Myocardial Infarction. 9, 16, 17

PUD Peptic Ulcer Disease. 16, 17

PVD Peripheral Vascular Disease. 17, 18

SAIL Secure Anonymised Information Linkage. 5, 7, 9, 19

TRE trusted research environment. 9

WIMD Welsh Index of Multiple Deprivation. 5, 21

WMC Wales Multimoribidity e-Cohort. 5

References

[1] Anna Cassell, Duncan Edwards, Amelia Harshfield, Kirsty Rhodes, James Brimicombe, Rupert Payne, and Simon Griffin. The epidemiology of multimorbidity in primary care: a retrospective cohort study. *British Journal of General Practice*, 68(669):e245–e251, 2018. ISSN 0960-1643. doi:10.3399/bjgp18X695465. URL `https://bjgp.org/content/68/669/e245`.

[2] Andrew Kingston, Louise Robinson, Heather Booth, Martin Knapp, Carol Jagger, and for the MODEM project. Projections of multi-morbidity in the older population in England to 2035: estimates from the Population Ageing and Care Simulation (PACSim) model. *Age and Ageing*, 47(3):374–380, 01 2018. ISSN 0002-0729. doi:10.1093/ageing/afx201. URL `https://doi.org/10.1093/ageing/afx201`.

[3] Jamie Burke and Daniel Schofield. Transforming healthcare data with graph-based techniques using sail databank. *NHS England*, 2022.

[4] James Rafferty, Alan Watkins, Jane Lyons, Ronan A Lyons, Ashley Akbari, Niels Peek, Farideh Jalali-Najafabadi, Thamer Ba Dhafari, Alexander Pate, Glen P Martin, et al. Ranking sets of morbidities using hypergraph centrality. *Journal of Biomedical Informatics*, 122:103916, 2021.

[5] C. Berge. *Graphs and Hypergraphs*. Mathematical Studies. North-Holland Publishing Company, 1976. ISBN 9780720404791. URL `https://books.google.co.uk/books?id=cXd2tQAACAAJ`.

[6] Jane Lyons, Ashley Akbari, Utkarsh Agrawal, Gill Harper, Amaya Azcoaga-Lorenzo, Rowena Bailey, James Rafferty, Alan Watkins, Richard Fry, Colin McCowan, Carol Dezateux, John P Robson, Niels Peek, Chris Holmes, Spiros Denaxas, Rhiannon Owen, Keith R Abrams, Ann John, Dermot O'Reilly, Sylvia Richardson, Marlous Hall, Chris P Gale, Jan Davies, Chris Davies, Lynsey Cross, John Gallacher, James Chess, Anthony J Brookes, and Ronan A Lyons. Protocol for the development of the wales multimorbidity e-cohort (wmc): data sources and methods to construct a population-based research platform to investigate multimorbidity. *BMJ Open*, 11(1), 2021. ISSN 2044-6055. doi:10.1136/bmjopen-2020-047101. URL `https://bmjopen.bmj.com/content/11/1/e047101`.