8-2022

Efficient Network Domination for Life Science Applications

Stephen K. Grady
University of Tennessee, Knoxville, sgrady3@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Stephen K. Grady entitled "Efficient Network Domination for Life Science Applications." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Life Sciences.

Michael A. Langston, Major Professor

We have read this dissertation and recommend its acceptance:

Faisal N. Abu-Khzam, Tian Hong, David J. Icove

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Efficient Network Domination for Life Science Applications

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Stephen Kent Grady
August 2022

DEDICATION

Dedicated to my parents who always taught me to take a step back and ask why.

ACKNOWLEDGEMENTS

I would like to first and foremost thank my advisor Dr. Michael Langston for his guidance and patience. From him, I have learned many invaluable lessons and what it means to be a scientist. Only with his aid did I make it to the finish line. I would also like to thank Dr. Faisal Abu-Khzam for helping me discover the joy in algorithm development. I must express my gratitude to my dissertation committee: Dr. Faisal Abu-Khzam, Dr. Tian Hong, and Dr. David Icove. I would like to thank Dr. Albrecht von Arnim and the Genome Science and Technology program for all the opportunities in pursuing my curiosity that have been afforded to me. I must also extend my appreciation to those with which I collaborated for all their hard work and insights.

I would like to express my eternal gratitude to Ashleigh Depetro for being there in the toughest of times and keeping my spirits high. Next, I would like to thank my lab mates Ron and Brett Hagan, Yuping Lu, Carissa Bleker, Chen Cheng, Levente Dojcsak, Hunter Leef, and the late Charles Phillips for all their help, brainstorming, and listening to my crazy ideas. And finally, I must acknowledge all the friends I've made along the way including Pawat Pattarawat, David Eberius, Thananon "Arm" Patinyasakdikul, Alex Cope, Alfredo Blakeley-Ruiz, Jordan Bush, and Matt Entler and thank them for all the good times we had.

ABSTRACT

With the ever-increasing size of data available to researchers, traditional methods of analysis often cannot scale to match problems being studied. Often only a subset of variables may be utilized or studied further, motivating the need of techniques that can prioritize variable selection. This dissertation describes the development and application of graph theoretic techniques, particularly the notion of domination, for this purpose. In the first part of this dissertation, algorithms for vertex prioritization in the field of network controllability are studied. Here, the number of solutions to which a vertex belongs is used to classify said vertex and determine its suitability in controlling a network. Novel efficient scalable algorithms are developed and analyzed. Empirical tests demonstrate the improvement of these algorithms over those already established in the literature. The second part of this dissertation concerns the prioritization of genes for loss-of-function allele studies in mice. The International Mouse Phenotyping Consortium leads the initiative to develop a loss-of-function allele for each protein coding gene in the mouse genome. Only a small proportion of untested genes can be selected for further study. To address the need to prioritize genes, a generalizable data science strategy is developed. This strategy models genes as a gene-similarity graph, and from it selects subset that will be further characterized. Empirical tests demonstrate the method's utility over that of pseudorandom selection and less computationally demanding methods. Finally, part three addresses the important task of preprocessing in the context of noisy public health data. Many public health databases have been developed to collect, curate, and store a variety of environmental measurements. Idiosyncrasies in these measurements, however, introduce noise to data found in these databases in several ways including missing, incorrect, outlying, and incompatible data. Beyond noisy data, multiple measurements of similar variables can introduce problems of multicollinearity. Domination is again employed in a novel graph method to handle autocorrelation. Empirical results using the Public Health Exposome dataset are reported. Together these three parts demonstrate the

utility of subset selection via domination when applied to a multitude of data sources from a variety of disciplines in the life sciences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

In this dissertation in parts we present work from three projects that share the notion of domination in order to reduce problems for better study. The data and applications we study come from the fields of biological network analysis, statistical genetics, and public health. All three fields have seen an explosion in the size and number of data available [1, 2]. These data are often modeled as a graph, sometimes referred to as a network, which is an ordered pair $G = \langle V, E \rangle$ that consists of a set of vertices $V(G)$ and a set of edges $E(G)$, or simply V and E, respectively.

The applications we study in this dissertation require a subset of vertices from these graphs. These subsets are often required to comprise a minimum number of elements, and that members of the selected subset cover all other elements in some manner. In the field of biological network analysis such subsets are required to interact and influence all elements in a graph while in statistical genetics and public health the selected subset should share similarity with and represent all elements in a graph. Given these constraints, we chose to study minimum dominating set and its variants in these respective contexts.

A minimum dominating set (MDS) is a set of vertices D such that all vertices in a graph G are either in D or adjacent to a vertex in D, where D is of smallest size. A minimal dominating set is a dominating set that cannot be made smaller. The cardinality of an MDS is denoted by $\gamma(G)$. MDS is both a classic *NP*-complete [3] and W[2]-complete [4] problem.

MDS has found a wide variety of uses in domains including network science [5-8], sensor placement [9], and transportation streaming [10]. In the field of systems biology MDS has been used to model the controllability of biological

networks in research fields such as cancer [11-13], drug discovery [14], gene regulation [15], neuroscience [16], protein interaction [17-19], viral infection [20], and ncRNA's latent regulatory role in polygenic human disease [21].

In Chapter 2, we begin with the notion of controllability in biological networks. This problem is modeled as classifying vertices based on the number of solutions to which they belong. In the work presented there, we developed and analyzed two novel efficient algorithms for this purpose that greatly improve upon existing techniques found in the literature. In Chapter 3, we concern ourselves with the prioritization of genes for loss-of-function allele production. This problem is particularly important to the International Mouse Phenotyping Consortium [22] that would like to expand its catalogue of loss-of-function allele models but is constrained in the resources available to do so. To address this problem, we applied an MDS solver to select a subset of genes of appropriate size that will deliver a diverse and representative set of loss-of-function models to maximize knowledge gain. In Chapter 4, we turn our focus to the domain of public health and describe aspects of noise and multicollinearity in their datasets as well as present a set of tools to address these problems. We also introduce a novel graph theoretic technique that employs MDS to reduce the prevalence of autocorrelates in a dataset. Finally in Chapter 5, we state concluding remarks as well as summarize contributions.

A Few Relevant Graph Theoretical Basics

When modeling data as a graph, vertices represent entities and edges between vertices represent a measure of interaction, similarity, relationship, etc. The selection of this measure is problem dependent. Simple metrics such as physical proximity or interaction are intuitive when data is derived from amenable sources. Mathematical notions of similarity may also be employed such as Pearson’s or Spearman’s correlation coefficient, or mutual information. As we will show in

Chapter 3, more abstract notions of similarity such as the probability of similarity based on random walks on graphs may also be used.

Given a similarity metric between vertices in a graph, a threshold is used to determine if an edge will be placed between them. The selection of a threshold is entirely problem dependent and can range from simple trial and error in conjunction with domain expertise to more advanced techniques such as spectral methods [23].

Previous Work

As this is a dissertation in parts and the three subsequent chapters are disjoint in their problem setting, each chapter will contain its own previous work and introduction section.

CHAPTER II

DOMINATION BASED CLASSIFICATION ALGORITHMS FOR THE CONTROLLABILITY ANALYSIS OF BIOLOGICAL INTERACTION NETWORKS

This chapter appears in a manuscript [24] of the same title by Stephen K. Grady, Faisal N. Abu-Khzam, Ronald D. Hagan, Hesam Shams, and Michael A. Langston published at *Scientific Reports*. My contributions include algorithm and implementation development, data collection and efficiency testing.

Abstract

Minimum dominating set is a classic *NP-Complete* problem that has found increasing use in a systems biology application setting. It is commonly used to classify vertices in the context of the number of solutions to which they belong. This can be useful to identify key vertices in biological data derived from RNA, protein interactions, or metabolic interactions among other sources. Current methods may have to solve an instance for each vertex in a graph, rendering them computationally prohibitive. To address this setback, two new classification algorithms are derived and tested for efficiency. Timings on real-world biological networks are reported.

1. Introduction

A graph G may have as many as $15^{n/6}$ distinct MDS solutions [25]. This upper bound makes the enumeration of all MDS solutions infeasible. A common strategy is therefore to concentrate on significance and classify a vertex as “essential” if it is used in every MDS, as “intermittent” if it is used in some but not every MDS, and as “redundant” if it is never used in any MDS.

Previous classification strategies examine vertices one by one, and thus invoke an MDS algorithm n or more times in the worst case. Efficiency may be achieved in the average case, however, by observing that a vertex is essential should it have two or more pendant vertices [26] and redundant should all of its neighbors be essential [27]. In this chapter we generalize and greatly extend these observations with five novel vertex classification rules with which we can further decrease the number of times MDS must be solved. To accomplish this, we

devised highly efficient techniques that can take advantage of neighborhood structure and, if desired, adjacency-preserving vertex permutations. Using these rules, we developed two classification algorithms with which we conducted a series of experiments on graphs derived from data sourced from a variety of biological application domain.

2. Preliminaries

2.1 Notation

Let u and v denote elements of V. The distance between u and v is the number of edges in a shortest path between them. The neighborhood of u, denoted by $N[u]$, comprises u and its neighbors or, equivalently, those vertices within distance one from u. (This is sometimes called the closed neighborhood of u, in order to distinguish it from the open neighborhood $N[u] - \{u\}$.) Neighborhoods can be extended to sets such that for a set of vertices S, the closed neighborhood of $N[S]$ denotes S and all neighbors of its elements. An orbit is an equivalency class of a vertex set under the action of an automorphism group. Stated another way, u and v belong to the same orbit if and only if there exists a relabeling of V that results in an isomorphic graph for which u and v have exchanged labels [28]. Given an MDS D, we say that u dominates v if u and v are adjacent and u but not v is an element of D.

2.2 Prior Work

The vertex classification problem has been studied [26, 27] using the previously mentioned observations coupled with an MDS algorithm that employs an Integer Linear Programming (ILP) solver. Once an initial MDS, D, has been computed, each vertex u is considered in turn:

- If $u \in D$, then construct an ILP instance of MDS with a constraint to exclude u. We refer to the resultant procedure as ILP-exclude, with parameters G and u. If γ(ILP-exclude(G,u)) exceeds $\gamma(G)$, then u is

essential, otherwise it is intermittent.

- And if $u \notin D$, then construct an ILP instance of MDS with a constraint to include u. We refer to the resultant procedure as ILP-include, also with parameters G and u. If γ(ILP-include(G,u)) exceeds $\gamma(G)$, then u is redundant, otherwise it is intermittent.

2.3 Classifier A

This previously unnamed procedure is presented here in pseudocode and dubbed Classifier A. Note that the exploitation of pendant vertices can be used before an initial MDS is computed, while the examination of neighbors is best applied only after all essential vertices have been identified.

```
Classifier A
input: A finite simple graph G=<V,E>
output: A partitioning of V into essential (aka critical) vertices C,
        intermittent vertices I, and redundant vertices R
begin
C := those elements of V with two or more pendant vertices
I := ∅
D := MDS(G)
for each unclassified u ∈ D
    if γ(ILP-exclude(G,u)) > γ(G)
        then C := C ∪ {u}
        else  I := I ∪ {u}
R := those vertices adjacent only to elements of C
for each vertex u still without a classification
    if γ(ILP-include(G,u)) > γ(G)
        then R := R ∪ {u}
        else  I := I ∪ {u}
end
```

Classifier A requires low-order polynomial time to initialize C and R, exponential time to call an ILP solver to answer a single instance of MDS and time for at most n exponential-time calls to ILP-exclude/include. Classifier A's needs for extra space are negligible.

3. Improved Classifiers

3.1 Classification Rules

The most time-consuming operations of Classifier A are its multitude of calls to ILP-exclude/include. Therefore, we propose, scrutinize, and employ a series of preprocessing rules in an effort to minimize the number of these calls.

Rule 1. Suppose u and v are adjacent, and the neighborhood of *u* is a proper subset of the neighborhood of *v*. If *v* is essential, then *u* is redundant.

Soundness. If an MDS contains *v*, then it cannot contain *u*, since otherwise the MDS would not be minimum. Thus, if every MDS contains *v*, then none can contain *u*. (Note the need for proper containment. If N[*u*] = N[*v*], then neither *u* nor *v* can be essential, and both must be redundant or both intermittent.)

Rule 2. If *u* is not essential, and if every element in *u*'s neighborhood is either essential or adjacent to an essential vertex, then *u* is redundant.

Soundness. This is a generalization of Rule 1, in which vertices in the neighborhood of *u* may be dominated by more than just a single essential vertex.

Rule 3. Suppose *u* but not *v* is contained in an MDS for which those vertices dominated only by *u* are in the neighborhood of *v*. Then both *u* and *v* are intermittent.

Soundness. Replacing *u* with *v* produces a distinct but equivalent MDS.

Rule 4. If *u* has neighbors *v* and *w* whose only common neighbor is *u* and for which (N[N[*v*]] ∪ N[N[*w*]]) ⊂ N[*u*], then *u* is essential.

Soundness. Because N[v] ∩ N[w] = {u}, and because u dominates every vertex in N[N[v]] ∪ N[N[w]], it follows that u is required in any MDS, since otherwise at least two vertices from N[v] ∪ N[w] would be required in its place to dominate v and w.

3.2 Classifier B

We make use of Rules 1-4 in a procedure named Classifier B. This new classifier need not invoke Classifier A as the observations on which Classifier A relies are subsumed by Rules 2 and 4. The order in which rule are applied by Classifier B is important to minimize the number of times ILP-include/exclude is invoked.

Classifier B
input: A finite simple graph $G=<V,E>$
output: A partitioning of V into essential vertices C, intermittent vertices I, and redundant vertices R
begin
C := the set of essential vertices found by Rule 4
$I := R := \emptyset$
for each vertex $u \in C$
    $R := R \cup$ all redundant vertices in $N(u)$ found by Rule 1
$D := MDS(G)$
I := all intermittent vertices found by Rule 3
for each unclassified vertex $u \in D-I$
    if γ(ILP-exclude(G, u)) > $|D|$
        then
            $C := C \cup \{u\}$;
            $R := R \cup$ all redundant vertices in $N(u)$ found by Rule 1
        else $I := I \cup \{u\}$
$R := R \cup$ all redundant vertices found by Rule 2
for each vertex u still without a classification
    if γ(ILP-include(G, u)) > $\gamma(G)$
        then $R := R \cup \{u\}$
        else $I := I \cup \{u\}$
end

Classifier B's resource requirements are similar to those of Classifier A. It needs low-order polynomial time to apply Rules 1-4 in the computation of *C*, *I*, and *R*. An exact upper bound is dependent on graph density and data structures used. It needs exponential time for an initial call to an ILP solver to answer a single instance of MDS, and time for at most *n* exponential-time calls to ILP-include/exclude. Classifier B's needs for extra space are negligible.

4. The Use of Algebraic Symmetry

4.1 Orbits and Automorphisms

To provide additional reductions in the number of calls to ILP-include/exclude we used notions of graph structure, neighborhood symmetry, and adjacency-preserving vertex permutations.

Rule 5. If *V* is partitioned into a set of vertex orbits, then vertices within the same orbit must possess the same classification.

Soundness. Vertices within the same orbit are indistinguishable under automorphic transformation, and so their classifications will be identical.

4.2 Classifier C

With the addition of Rule 5, we produced a third procedure, which we christen Classifier C. This classifier works much as does Classifier B with the exception that it incorporates Rule 5 by first computing all orbits and then, whenever a vertex is classified, any unclassified vertices in its orbit are assigned the same classification.

Classifier C, like Classifier B, requires low-order polynomial time to apply Rules 1-4, exponential time to solve a single instance of MDS, and time for at most *n* exponential-time calls to ILP-exclude/include. Classifier C also needs low-order polynomial time to update orbit classifications. More significantly, it requires

exponential time to determine the orbits themselves with known practical methods [29]. These orbits can be found using bliss [30], nauty [31], and a variety of other popular, well documented, easy-to-use tools. From these we chose saucy [32, 33], by virtue of the fact that it has been tuned for sparse graphs, which are overwhelmingly representative of large-scale biological data. And indeed, saucy was roughly 10-20 times faster than bliss and over 1000 times faster than nauty across our test suite. We hasten to add, however, that saucy requires a bit more effort to implement than does nauty or bliss. This is because saucy only returns vertex pairs that occupy the same orbit. The user must then merge these pairs to form a complete orbit set. Classifier C's needs for extra space are negligible.

5. Classifiers Comparisons

5.1 Computational Milieu

Classifiers A, B, and C were implemented in C++ and compiled using the g++ (GCC) version 4.8.5 compiler under the CentOS Linux 7 x86-64 operating system. Various mathematical optimization software packages were considered, including notable options such as CPLEX [34] and Xpress [33]. From these we chose Gurobi [34] for our ILP solver. It is a hugely successful, widely used, state-of-the-art commercial product. Moreover, Gurobi is freely available to many in the research community via an academic site license. As in previous work, we used ILP to satisfy each classifier's initial MDS requirement. Possible alternatives include the measure and conquer method of [35], which runs in $O(1.4864^n)$ time and polynomial space. We were careful to avoid reproducibility problems that might arise from complex parameter settings. Our classifiers take as input only finite simple graphs, while default settings were strictly obeyed for Gurobi.

Three dozen challenging graphs were assembled to form a comprehensive classifier test suite. Graphs that populate this suite were obtained from well-known repositories and derived from transcriptomic, proteomic, epigenetic, and a

variety of other sorts of biological data. We excluded from this suite any graph on which one or more classifiers failed to finish within 24 hours, which generally seemed to result from exceptional size or, less frequently, from unusual density. Descriptions of each graph used in comparisons are in Table 1 (Appendix). Runtimes per instance and classifier are found in Table 2 (Appendix).

5.2 MDS calls comparisons

We measured the success of preprocessing as a percentage of vertices classified with an ILP-include/exclude call. Over our test suite, Classifier A had an average success rate of only 14.1%. Classifier B, in contrast, had an average success rate of 67.2%, while Classifier C had an average success rate of 72.5%. Such improvements place Classifiers B and C at an enormous computational advantage. Success percentages for each tested network can be found in Figure 1.

5.3 Runtime comparisons

Beyond preprocessing success rates, we tested if these preprocessing rates translated to improved runtimes. Indeed, we found that Classifier A was simply not competitive to the latter two classifiers. The difference in runtime between Classifiers B and C, however, deserves consideration with Classifier C's time-consuming orbits computations. Results were mixed. Leading-edge graph automorphism packages such as saucy, still struggle to compete with ILP computations performed by a well-honed commercial product such as Gurobi. Runtimes varied greatly, so for ease of comparisons, we normalized all runtimes to that of Classifier A. This revealed that Classifiers B and C performed, on average, roughly the same. Classifier B took approximately 38.2% as long as Classifier A, while Classifier C took some 37.9% as long. Therefore, given our test suite, the additional computational requirements of Rule 5 were barely noticeable. Figure 2 depicts both classifiers performance on each test network.

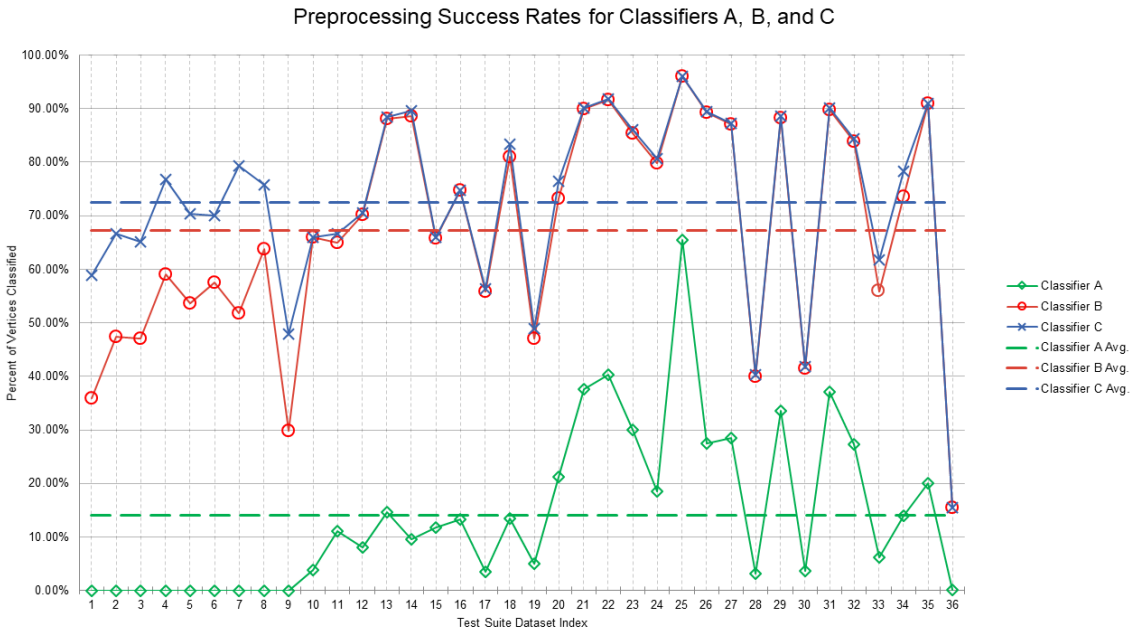


Figure 1: Percent of vertices classified without ILP-exclude/include *Percent of vertices classified without ILP-exclude/include calls by Classifiers A (in green), B (in red), and C (in blue). Dashed lines represent averages, which were 14.1%, 67.2%, and 72.5% for Classifiers A, B, and C, respectively.*

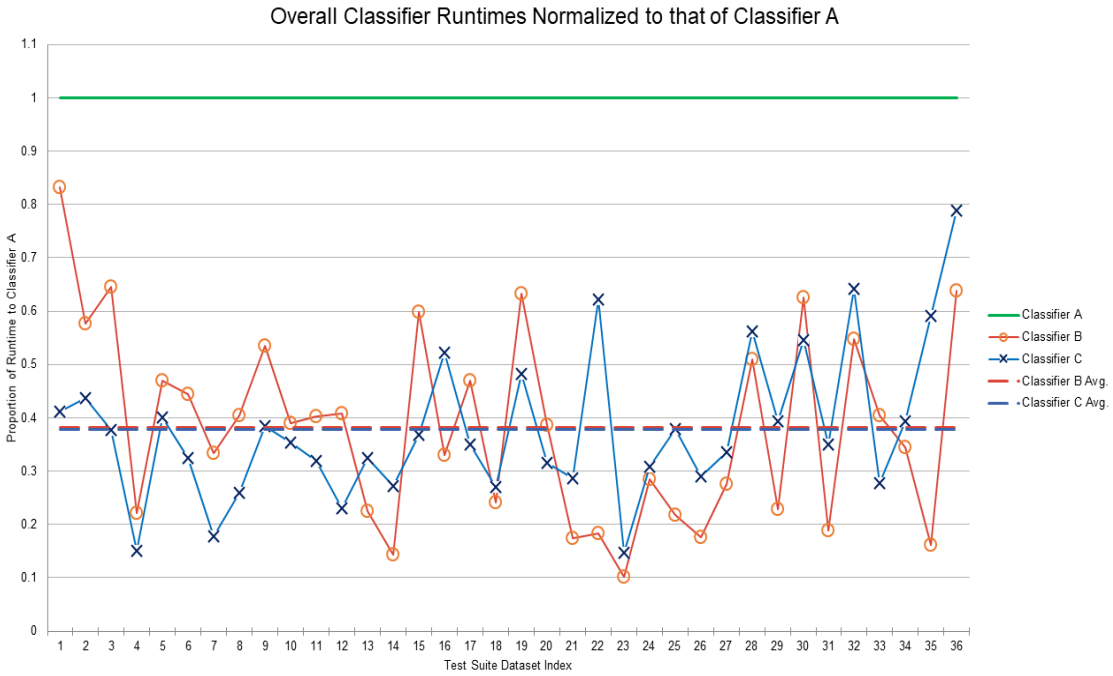


Figure 2: Overall Runtime of Classifiers *Overall runtimes of Classifiers B (in red) and C (in blue), normalized to that of Classifier A (in green). Dashed lines are almost collinear and represent averages, which were 38.2% and 37.9% for Classifiers B and C, respectively.*

6. Discussion

6.1 Conclusions

In this chapter we have developed, analyzed, implemented, and tested five novel classification rules and two highly innovative classifier algorithms with which vertex significance can be gauged in a network domination setting. Extensive empirical evidence of the practical usefulness of these powerful new rules and classifiers was also generated using a comprehensive test suite centering on life science applications and biological data.

Classifiers B and C turn out to be huge improvements over Classifier A in terms of both preprocessing rates and overall runtimes. Their relative effectiveness would have been even more pronounced had we not had access to a commercial ILP solver with the exceptional efficiency of Gurobi. Results from our extensive test suite suggest that Classifiers B and C are very nearly equal in performance. Although Classifier C was faster by a narrow margin, users may wish to give Classifier B a slight nod for its comparative simplicity.

Patterns seen in results and data may be of additional interest. We observe, for example, the modest MDS size of chromatin interaction data (test graphs 1-9). Concomitantly, these are the only graphs for which the preprocessing performed by Classifier C is significantly better than that of Classifier B. It seems plausible that this rather curious situation might be attributable to graph density, but most biological data is sparse, and indeed these graphs are roughly as sparse as all others in our test suite. We therefore turned to degree distributions and found that the chromatin interaction histograms appear normalesque and not scale-free like histograms for the rest of our test suite. Whether this is causative is unknown. We found it interesting too that all classifiers were unusually successful in preprocessing graph 25 (bio-grid-worm). Upon investigation, we discovered that this graph has an extremely high number of redundant vertices. Whether this

attribute relates to better preprocessing is unclear. And finally, graph 36 (bn-mouse-retina-1) caught our attention because it was especially difficult for all classifiers, and yet its MDS is about the same size as those of the chromatin interaction graphs. Other than idiosyncrasies of data capture (neuronal connections imaged by electron microscopy), we can posit no particular basis for its computational recalcitrance.

6.2 Directions for Future Research

The rules we have devised assign a single MDS classification to any vertex. It is sometimes possible, however, to eliminate one classification option, making it reasonable to envisage more convoluted rules that assign a pair of classification choices to some vertices. As we have seen with Rule 5, however, the overhead and complexity of such a strategy must not be so high that it negates any meaningful gains.

MDS vertex classifications may find additional utility among problem variants. The study of independent dominating set, for instance, is a restatement of maximal independent set, and can be traced back roughly 60 years [36]. Other classic examples include connected dominating set [37] and total dominating set [38]. Vertex classification strategies may also be of interest when data is drawn from reduced graph families. Limiting inputs to planar graphs, for example, is a popular restriction in circuit layout and many other engineering applications, although in our opinion this sort of limitation would be difficult to motivate from a biological perspective.

It might also be instructive to consider the relationship between orbit distributions and graph structure. For example, those who embrace the once-popular scale-free hypothesis [39] might predict that orbits would be found primarily among leaves that share a common neighbor. As a simple test, we therefore scanned the non-singleton orbit lists and computed the percentage of these lists that

contained non-leaf vertices for each graph in our test suite. These values turned out to range more or less uniformly between 4% and 100%. Unsurprisingly, it thus appears that the utility of automorphic transformation is highly data dependent, and that the extent to which Rule 5 applies is primarily a function of the particular graph under examination. This would seem to suggest that the relationship between orbits and the topology of graphs derived from biological data might warrant future study.

Finally, while our focus has been on practical applications, numerous theoretical questions beckon. We think it highly probable, for example, that classification strategies such as those we have developed here may prove useful for combinatorial problems other than MDS. Rule 5, in particular, seems to have something of a universal appeal. Another good example rests with worst-case classifier behavior. Each method we have considered could in principle invoke an MDS solver as many as n+1 times. Classifier A in fact did exactly this, for instance, on test graph 5 (HiC-Net-10). Classifiers B and C, on the other hand, never even came close to this sort of pathology. We think it is highly unlikely that real-world biological data of sufficient size would cause either of these classifiers to be so completely ineffective. To the best of our knowledge, however, the sort of worst-case performance that might be attained with highly contrived data remains unknown.

Appendix

Table 1: Test suite of real-world biological graphs. *Types are CI (chromatin interaction), GC (gene co-expression), GFA (gene functional association), PPI (protein-protein interaction), and M (miscellaneous), where graph 32 is derived from biological functionality data, graph 33 is derived from drug-drug interactions, graph 34 is derived from human gene signaling and regulatory pathway interactions, and graphs 35 and 36 are derived from neuron connections in the fly medulla and in the mouse retina, respectively.*

Index	Graph Name	Type	Source(s)	|V|	|E|	□
1	HiC-Net-1	CI	[40]	1099	32848	17
2	HiC-Net-3	CI	[40]	1084	31724	19
3	HiC-Net-5	CI	[40]	1419	43763	25
4	HiC-Net-7	CI	[40]	1083	32336	18
5	HiC-Net-10	CI	[40]	1094	30216	20
6	HiC-Net-11	CI	[40]	1165	38784	17
7	HiC-Net-14	CI	[40]	1056	33851	15
8	HiC-Net-15	CI	[40]	1164	35470	19
9	HiC-Net-21	CI	[40]	1376	41314	22
10	GIANT-top-brain-02-filtered	GC	[41]	14306	1358435	1159
11	Pancreas_GDS4102_control.995	GC	[42-45]	2591	61245	650
12	ProteomeHD-top-05-co-regulated	GC	[41]	2717	62749	505
13	ColorectalCancer_GSE9348_control.975	GC	[42, 46]	2803	3918	1099
14	BreastCancer_GSE10810_case	GC	[42, 47]	3249	7070	1197
15	ParkinsonsDisease_GSE20141_case.996	GC	[42, 48]	2340	12959	738
16	cerebellum-male	GC	[49]	10274	78981	2605
17	yeast-8	GC	[42, 50]	5544	389058	409
18	bio-CE-GT	GFA	[51]	924	3239	126
19	bio-CE-GN	GFA	[51]	2220	53683	195
20	Bio-HS-HT	GFA	[51]	2570	13691	456
21	BioGrid-PP-Interaction-A-thaliana	PPI	[41]	10823	51278	1353

Table 1 Continued

Index	Graph Name	Type	Source(s)	|V|	|E|	□
22	Y2H-union	PPI	[52]	1966	2705	575
23	bio-grid-fission-yeast	PPI	[51]	2026	12637	280
24	HC-BIOGRID-2.0.31	PPI	[53]	2538	6418	607
25	bio-grid-worm	PPI	[51]	3507	6531	578
26	HuRi	PPI	[54]	8275	52088	1341
27	bio-grid-fruitfly	PPI	[51]	7274	24894	1522
28	bio-wormnet-v3	PPI	[51]	16347	762822	2072
29	bio-grid-human	PPI	[51]	9436	31182	1785
30	PP-Decagon-ppi	PPI	[40]	19081	715612	1353
31	Lit-BM	PPI	[41]	5956	12758	1322
32	FF-miner-miner-func-func	M	[41]	46027	106510	6751
33	ChCh-Miner-drugbank-chem-chem	M	[51]	1514	48514	93
34	NCI-PID-complete-interactions	M	[51]	2855	25433	247
35	bn-fly-drosophila-medulla-1	M	[40]	1781	8911	317
36	bn-mouse-retina-1	M	[40]	1076	90811	14

Table 2: *Run times for each test suite instance and each classifier, measured in seconds.*

Index	Graph Name	Classifier A	Classifier B	Classifier C
1	HiC-Net-1	18.47	15.37	7.586
2	HiC-Net-3	15.054	8.689	6.571
3	HiC-Net-5	36.248	23.372	13.646
4	HiC-Net-7	30.824	6.833	4.65
5	HiC-Net-10	17.334	8.126	6.955
6	HiC-Net-11	19.584	8.698	6.342
7	HiC-Net-14	21.723	7.265	3.852
8	HiC-Net-15	16.313	6.589	4.231
9	HiC-Net-21	48.166	25.792	18.477
10	GIANT-top-brain-02-filtered	1.628	0.394	0.441
11	Pancreas_GDS4102_control.995	37.308	23.598	17.987
12	ProteomeHD-top-05-co-regulated	16.162	6.238	5.109
13	ColorectalCancer_GSE9348_control.975	9903.397	3854.583	3499.532
14	BreastCancer_GSE10810_case	42.488	17.14	13.543
15	ParkinsonsDisease_GSE20141_case.996	62.139	25.381	14.327
16	cerebellum-male	11.396	2.567	3.708
17	yeast-8	17.793	2.542	4.848
18	bio-CE-GT	45.532	27.24	16.747
19	bio-CE-GN	317.515	105.023	165.61
20	Bio-HS-HT	976.197	458.888	341.572
21	BioGrid-PP-Interaction-A-thaliana	151.596	26.463	43.407
22	Y2H-union	3.216	0.591	1.999
23	bio-grid-fission-yeast	11.455	1.684	1.684
24	HC-BIOGRID-2.0.31	11.074	3.146	3.417
25	bio-grid-worm	4.856	1.056	1.842
26	HuRi	115.966	20.366	33.67
27	bio-grid-fruitfly	80.533	22.184	27.024
28	bio-wormnet-v3	8533.929	4352.012	4791.11
29	bio-grid-human	118.688	27.055	46.685
30	PP-Decagon-ppi	10369.653	6483.164	5656.752
31	Lit-BM	37.214	7.03	13.032

Table 2 Continued.

Index	Graph Name	Classifier A	Classifier B	Classifier C
32	FF-miner-miner-func-func	21.408	7.394	8.435
33	ChCh-Miner-drugbank-chem-chem	42.936	27.369	33.892
34	NCI-PID-complete-interactions	5.955	0.961	3.518
35	bn-fly-drosophila-medulla-1	25.876	10.465	7.175
36	bn-mouse-retina-1	8728.462	3226.134	4704.256

CHAPTER III

A GRAPH-THEORETICAL APPROACH TO EXPERIMENT PRIORITIZATION IN GENOME-WIDE INVESTIGATIONS

This chapter is from a manuscript in preparation of the same title by Stephen K. Grady, Kevin A. Peterson, Stephen A. Murray, Erich J. Baker, Michael A. Langston, and Elissa J. Chesler. My contributions include algorithm development, implementation, and testing.

Abstract

High throughput investigations of biological systems generate large datasets. From these datasets, only a relatively small proportion of validation experiments may be performed. One such example is motivated by the International Mouse Phenotyping Consortium, which is a global effort to characterize all mouse orthologs of human genes through loss-of-function allele models. While this effort has to date generated such models for approximately 7,000 genes, out of the thousands left, only an additional 1,500 may be studied due to resource constraints. To aid in this selection, we developed an unbiased pipeline that modeled heterogenous biological data as a knowledge graph to which a minimum dominating set solver was applied to select a representative subset. Experiments on Gene Ontology retrieval demonstrated that minimum dominating set outperforms selection by pseudorandom selection and other less computationally intense methods.

1. Introduction

In systems biology experiments, data is often collected for every gene in the genome. Researchers have represented these data as experimentally derived networks, including gene-coexpression networks [55], Bayesian networks of genes and phenotypes [56], networks of temporal relations among genes [57], and many others. Given these networks, researchers typically must focus on a very limited subgraph, or even a single “hub” node, when performing experimental validation. It is often impractical, however, to evaluate the entire graph to confirm the estimated relations between nodes. Thus, methods are

needed to identify which experiments would provide information about relations across the entire graph in a resource effective and efficient manner.

Increasingly, omics data is being incorporated in knowledge graphs that represent biological similarity and other relationships such as interacting members of biological networks. Examples include knowledge graphs for gene function similarity [58], phenotype similarity [59], disease similarity [60], and genotype-phenotype associations with disease [61]. With knowledge graphs such as these, information pertaining to an entity (vertex) can be applied in various "guilt-by-association" algorithms to infer relational knowledge about its neighbors [62]. It is widely appreciated, however, that such graphs are often sparse, or that there are substantial disparities among elements in their extent of characterization. In research on gene function, this disparity in knowledge has led to the concept of the ignorome [63, 64], the set of understudied and non-studied genes. The factors leading to these disparities in knowledge are multifaceted, including technological constraints, reagent availability, and the propensity of researchers to give further attention to previously studied genes, which may already be plausibly associated with a disease in question [65]. The ignorome deserves examination not only for a comprehensive understanding of biological systems, but also for the fact that such a large number of poorly characterized genes can leave researchers and medical professionals in the dark during critical moments such as the recent COVID-19 pandemic [66].

Characterizing the ignorome is of particular interest to the International Mouse Phenotyping Consortium (IMPC), which aims to produce a loss-of-function (null) allele for every protein coding gene and characterize each with a standardized phenotyping pipeline aimed to improve the breadth of knowledge of gene function in disease related traits. To date, the IMPC has generated and analyzed null alleles for 7,824 protein-coding genes out of the approximately 17,000 orthologous protein-coding genes shared between mouse and human [22].

Despite these efforts and technological advances such as CRISPR/Cas9 that help facilitate the production of null alleles, a significant proportion of the mouse genome remains to be studied.

Given both the feasibility of specific gene perturbations, in addition to time and resource constraints, the next phase of the project will need to prioritize a selection of 1,500 genes from the remaining untested set. The IMPC typically selects genes by committee, specific interest, or database mining such as gnomAD which focuses on functional constraints in the human genome [67]. Moving forward, the IMPC would benefit from a systematic method to select a diverse and informative set of genes of proper size to utilize their resources most efficiently for the purpose of increasing the breadth of knowledge of gene function.

In this chapter, we present a rational system for selecting a subset of genes from a biological knowledge graph that maximizes similarity and thus potential knowledge gain. We formulated the problem of selecting genes as that of computing an MDS from a gene-similarity knowledge graph. MDS is particularly advantageous in that it will sample all parts of the knowledge graph to deliver a diverse set of entities that share some similarity with the whole while also minimizing the number of entities chosen, thereby providing a tractable set for experimentation. Further, simple bootstrapping reveals that the knowledge graph used in this work and others like it conform to a scale-free (power-law) distribution. Thus, the results of [6] indicate that an MDS for it should be quite manageable relative to, say, an MDS for a pseudo-random graph of the same order. In addition, we developed a vertex weighting scheme using domain expert input to guide MDS selection. Using this overall strategy, we selected a set of 1,513 genes that can be further prioritized for null allele production by large-scale efforts such as the IMPC. This work is generalizable as MDS can be extended to

any systems biology study where network structure can be leveraged to ensure uniform sampling in cases where it is not feasible to test all elements.

2. Methods

2.1 Gene-Similarity Graph Construction

We first constructed a gene-similarity knowledge graph in which vertices represented genes, and edges between vertices were weighted by the network enhanced similarity search (NESS) [68] algorithm to interrogate mouse-centric heterogenous graphs constructed from Gene Ontology (GO) [69, 70], GeneWeaver [71], and String [72]. NESS applied a random walk with restart to constructed networks using a restart parameter of 0.35 as previously described [68]. Each seed was iteratively visited and its affinity to local and global genes were determined through a whole graph traversal until a convergence threshold of 10^{-8} was reached. The output denotes the probability of visiting a gene from a starting seed. Probabilities between each pair of genes were normalized and assigned as edge weights. This process was carried out for 16,897 protein coding genes with high confidence human orthologs and resulted in a fully connected edge weighted gene-similarity graph.

2.2 Integration of External Prioritization Information

To allow for the incorporation of domain expert knowledge and to guarantee the production of previously unavailable mouse resources, we incorporated a weighting scheme into the MDS algorithm that considers prior work for a given gene. Weights were generated using the number of known null allele counts for each gene and assigned to their corresponding vertex. This ensured the selection of understudied genes as opposed to genes that already have an existing loss-of-function allele. The number of distinct loss-of-function allele genes in our gene-similarity graph was obtained from MouseMine [73]. Custom queries were constructed to filter for allele type, attribute (contains null), and transmission to account for alleles generated using either embryonic stem cells

(targeted) or CRISPR/Cas9 (endonuclease mediated). Null allele types not accounted for in this query include those generated by other forms of random mutagenesis (e.g. spontaneous, gene trap and ENU mutagenesis) as well as conditional-ready alleles without a germline null reported.

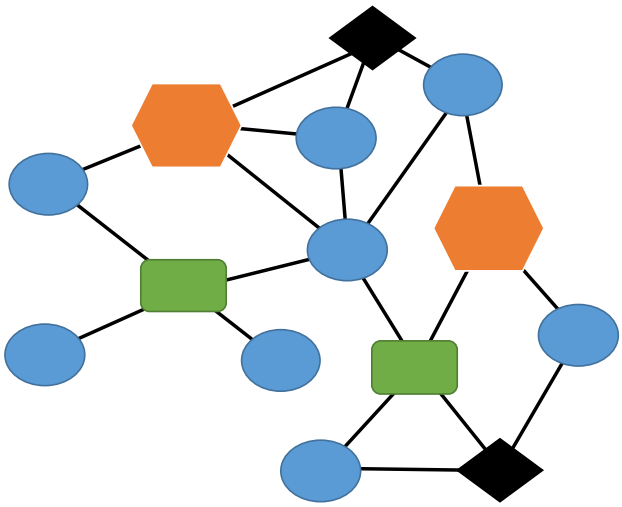
2.3 Gene Set Selection

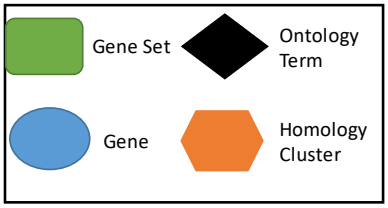
We applied MDS to the gene-similarity graph to select a subset of genes for null allele production. For extreme efficiency, we formulated MDS as the standard ILP formulation with an added extra constraint for those vertices, *R*, known in advance to be excluded from any solution. Vertices in *R* corresponded to genes with more null alleles than a given threshold or that had a null allele generated by the IMPC. The ILP formulation was therefore:

$$\text{minimize } \sum x_i \; \forall \; i \in V \text{ subject to } \sum x_i \geq 1 \; \forall \; i \in \mathrm{N}[v] \text{ and } \sum x_i < 1 \; \forall \; i \in R$$
$$\text{where } x \in \{0,1\}$$

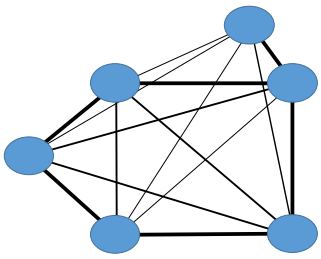
The size of an MDS is of course dependent on the size, density, and topology of an input graph. To select approximately 1,500 genes, we produced unweighted graphs using thresholds ranging from 0.01 to 0.99 in increments of 0.01, where an edge was retained if and only if its weight was at least as great as a given threshold. For each unweighted graph, we performed the following three steps. First, we added to *R* all genes that had an IMPC generated null allele or that had more than two null alleles, and we removed from *E* any edge between two vertices in *R*. Second, with the modified graph as input we computed an MDS with the above ILP formulation. And third, after an MDS was selected, we utilized Rule 3 from Chapter 1 to determine possible substitutes to genes in the selected MDS. This was done in case a selected gene leads to a non-viable null allele. After an MDS was computed for each threshold, one of size closest to 1,500 was retained for further study and model generation. Figure 3 demonstrates a step-by-step example of our method.

Figure 3: *Determining an MDS from a heterogenous knowledge graph.* ***A:*** *Depiction of a heterogenous knowledge graph incorporating diverse biological resources.* ***B:*** *Construction of an edge-weighted gene similarity graph via the NESS algorithm. Edge weights are depicted by line thickness.* ***C:*** *An unweighted graph generated by thresholding.* ***D:*** *Vertex weights determined by known null allele counts.* ***E:*** *An MDS (shown in red) is selected*

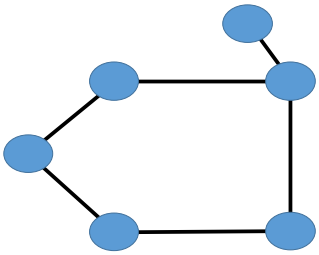



Gene Set
Ontology Term
Gene
Homology Cluster

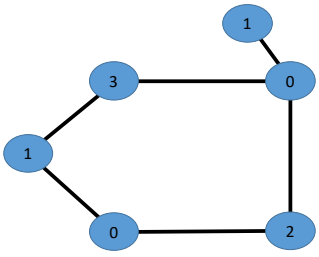
A



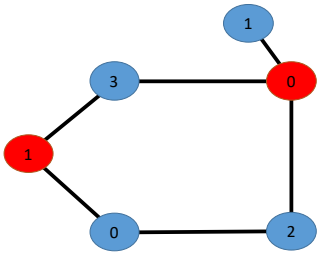
B



C


1
3
0
1
0
2

D


1
3
0
1
0
2

E

2.4 Utility Verification

To gauge the utility of MDS in improving the extent of knowledge coverage associated with a given set of genes, we compared Gene Ontology (GO) annotations for members of gene sets produced by MDS to the annotations of pseudorandomly selected genes. We computed the Jaccard similarity [74] between GO terms associated with genes in a selected subset and its complement. We refer to this metric as the “representative proportion” which measures the degree to which a selected subset shares biological similarity with its complement. Stated another way, of the GO terms that can be generated from a set of genes, the “representative proportion” shows what proportion can be obtained with a particular subset. Genes represented by vertices in our graphs often lack high quality characterizations. We therefore focused on genes for which there are IMPC generated null allele models and extracted subgraphs containing only vertices denoting these genes from our original gene-similarity graphs. We then subjected them to edge-weight thresholds from 0.40 to 0.95 at steps of 0.05 and computed an MDS for each connected component of the resultant graphs, with *R* being left empty. A pseudorandom set of genes of matching cardinality was selected at every iteration. Representative proportions were then measured. At each threshold, the process was repeated ten times for MDS, constantly shuffling the vertex order to toggle any tie breakers that might occur. Pseudorandom selection is vastly faster, and so we repeated the experiment 100 times at each threshold for it. Genes were pseudorandomly selected by first selecting all genes corresponding to vertices in the connected components of a graph. From this set, a subset was selected using the python *random [75]* package that is based on the “Mersenne Twister” pseudorandom number generator [76].

3. Results

3.1 Selected Gene Set

We set the null allele count threshold to two and computed MDS across an array of edge-weight thresholds as described in the previous section. A null allele threshold of two was selected to remedy any differences in previous model outcomes. The MDS of size closest to 1,500 was found at threshold 0.08. The vertices in this MDS corresponded to 1,513 genes, 1,370 of which do not have a current null allele. Of the selected genes, 50 existed as singletons, vertices with no neighbors, and are of particular interest for model production as they represent vertices for which there is little connectivity in our gene-similarity graph and may have little to no current information. Unfortunately, for the selected singletons this lack of information is mostly prescient as they were largely comprised of olfactory receptors (n = 27) which have largely been understudied due to phenotyping challenges and lack of cross-species homology. Non-singleton genes were found to be involved in a range of biological processes, with RNA processing (p-value = 1.75×10^{-4}) and ribosome biogenesis (p-value = 0.014) found to be significantly enriched with an FDR < 0.05 (Holm-Bonferroni Corrected). These findings are consistent with recent analysis highlighting factors involved in core biological processes that comprise a large fraction of genes lacking a null allele [77]. No significant enrichment was observed for other GO categories. In total, these findings provide supporting evidence that our strategy selected a diverse set of genes covering a broad range of biological processes. Understanding their function will enhance our current knowledge with potential to highlight novel connections between currently unrelated components.

We also performed a comparison of subgraph domination by MDS, domain expert nominations [78], and pseudorandom selection. Coverage varies dramatically based on method. The MDS, by definition, dominates the entire subgraph, outperforming the other methods. A visual comparison of the coverage

of the subgraph from which an MDS was selected can be found in Figure 4. The methods by which domination was determined were by MDS, domain expert nominations, and pseudorandom selection. The subgraph consisted of 360 vertices. Five MDS vertices dominated the entire subgraph and can be found in Figure 4.A. The domination of the subgraph by a pseudorandom selection of five vertices can be found in Figure 4.B. To further visually compare subgraph domination the vertices corresponding to the 60 genes with an existing null allele model, representing the collective work of domain experts is depicted in Figure 4.C. Interestingly, this set's corresponding 60 vertices only dominated 67.5% of the subgraph. This may be due to the feedback loop in which only genes similar to previously studied genes are further studied, creating a clustering phenomenon of null allele models. For completeness of comparisons, the domination of the subgraph by 60 pseudorandomly selected vertices can be found in Figure 4.D. These 60 vertices dominated a comparable amount of the graph as the domain expert selected genes at 71.7% of vertices.

3.2 Evaluation of Information Capture by MDS.

MDS produced higher representative proportions of term annotations than pseudorandom selection at every tested threshold (Figure 5). As can be seen in Figures 4.B and 4.D, vertices selected pseudorandomly are often found in clusters. The addition of selecting more vertices in a cluster does not bolster that subset's representative proportion. MDS by guaranteeing total graph domination is able to sample from all parts of our gene-similarity knowledge graph.

3.3 Evaluation of Minimal vs. Minimum Dominating Set

The results of our utility verification test are illustrated in Figure 5. MDS produced higher representative proportions than pseudorandom selection at every tested threshold. Note that MDS is a global optimization metric. Because it is *NP*-hard [79] and thus highly demanding, we sought to determine whether its

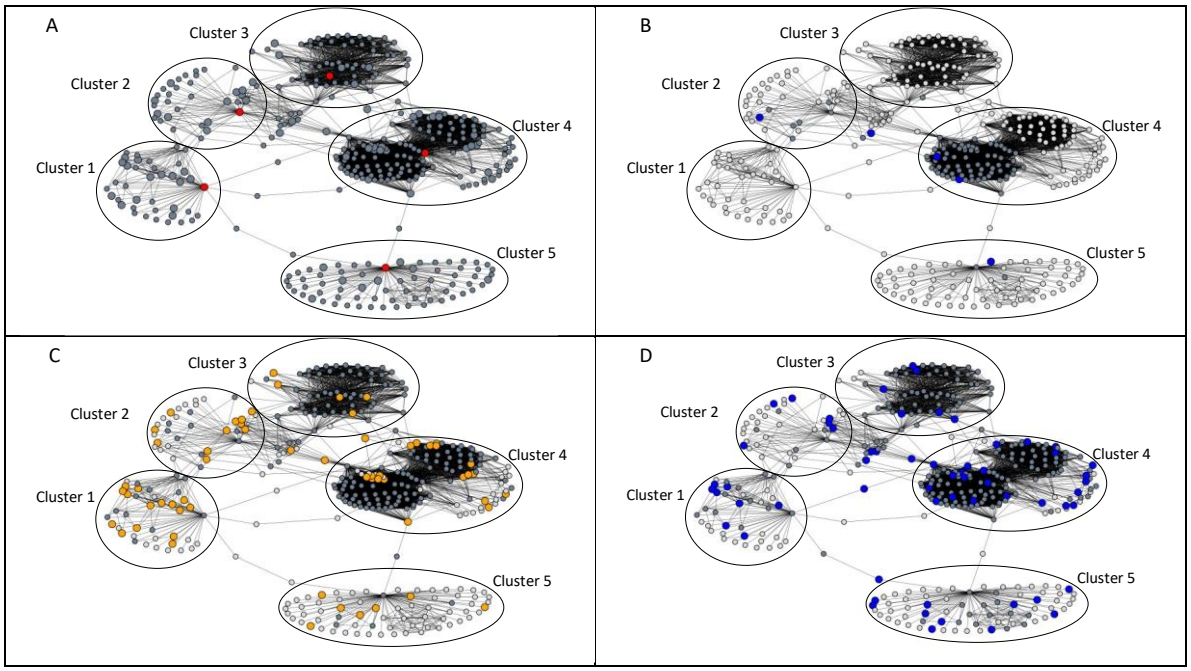


Figure 4: *Comparison of vertex domination by MDS, domain experts, and pseudorandom selection on a subgraph of 360 vertices from the gene-similarity graph at an edge-weight threshold of 0.08. The subgraph was extracted by selecting the neighborhoods of five vertices in the selected MDS. For each depiction, dominated vertices are in dark gray while non-dominated vertices are in white.* ***A.*** *Five vertices from the MDS, depicted in red, dominate all 360 vertices of the subgraph.* ***B.*** *A set of five pseudorandomly selected vertices are depicted in blue. This set dominates 84 or 23.3% of the subgraph. Note the complete or nearly complete loss of domination in all clusters with the exception of Cluster 4.* ***C.*** *A set of 60 vertices corresponding to genes that have a null allele generated by both the IMPC and wider community [78] depicted in orange. This set dominates 243 vertices or 67.5% of the subgraph which is most of the subgraph, but notably lacks domination in Clusters 1, 2, and 5.* ***D.*** *A set of 60 vertices selected pseudorandomly depicted in blue. This set dominates 258 or 71.7% of the subgraph. Note that this set's domination is roughly equivalent of that in C.*

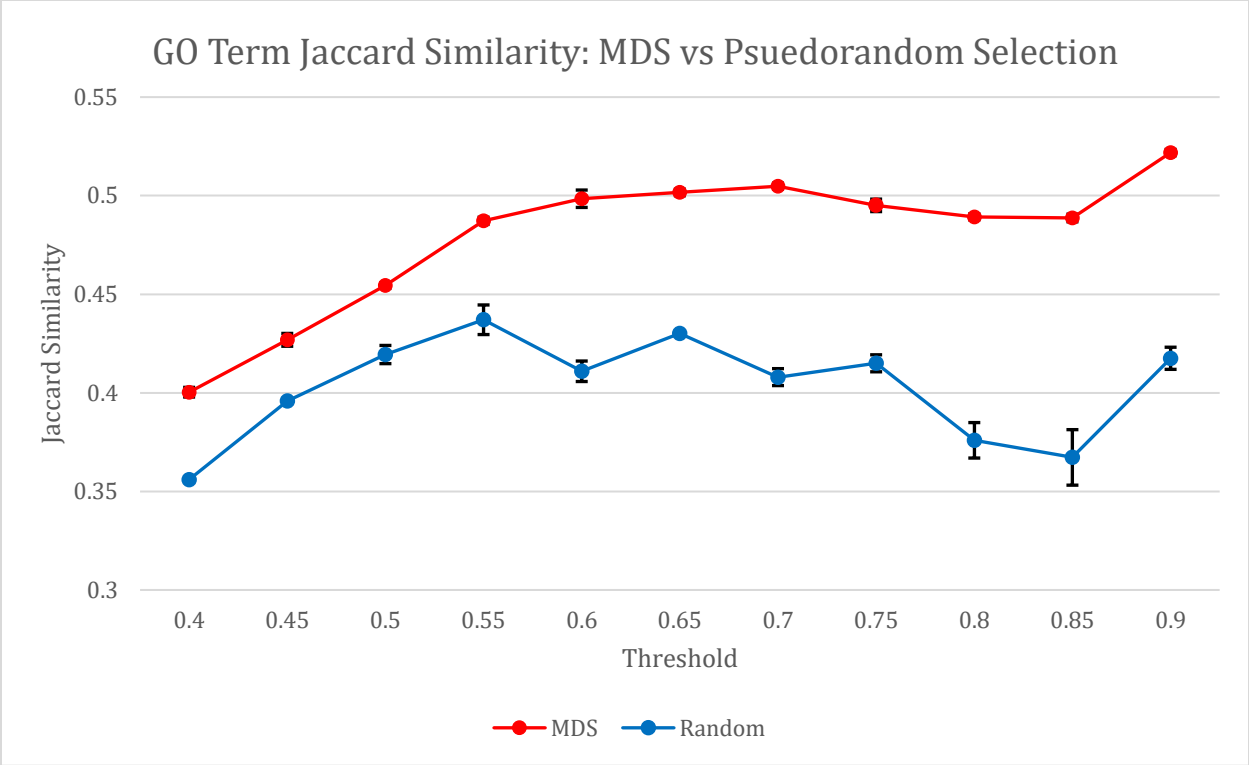


Figure 5: *GO term coverage tests for IMPC genes. MDS and pseudorandom selection were compared using Jaccard similarity scores between GO terms for genes in a subset by the method versus scores for genes in its complement. This test was repeated ten times for MDS and 100 times for pseudorandom selection, using thresholds from 0.40 to 0.95 in increments of 0.05. MDS had higher similarity scores than did pseudorandom selection across all thresholds.*

computational recalcitrance can be circumvented with minimality relaxation. We therefore computed minimal dominating sets as well. Such a set is simply one that cannot be made smaller by the deletion of a single vertex, making it a mere local optimization rule and one that needs but time linear in |*E*|. A greedy method such as that found in [80], for example, will suffice by iteratively selecting a vertex with the most undominated neighbors until a dominating set is found. As shown in Figure 6, however, minimality relaxation performed even worse than that of pseudorandom selection. Thus, it seems that the exhaustive computational demands of MDS are warranted.

4 Discussion

4.1 Conclusions

We have demonstrated the development, implementation, and use case of a novel, high-performance novel strategy to select a subset of genes for inferring global gene functional knowledge while maximizing the use of experimental resources use. Using a set of curated genes by the IMPC, we demonstrated our method's ability to select a gene set that represents covered genes better than pseudorandom selection and the simpler computation of minimal dominating set. Our method outperformed both, demonstrating its utility for studies seeking to maximize knowledge gained across the breadth of a biological network. We note this latter result particularly interesting as it provides evidence that the extra computation that goes into solving MDS pays off in generating a subset of genes with fewer overlapping GO terms. This is the first use case of MDS being applied to guide experimental selection known to the authors.

Towards the goal of diminishing the number of genes included in the ignorome, genes identified by MDS displayed promising properties for new knowledge generation. First, most of the selected genes have not been included in loss-of-function studies in the mouse.

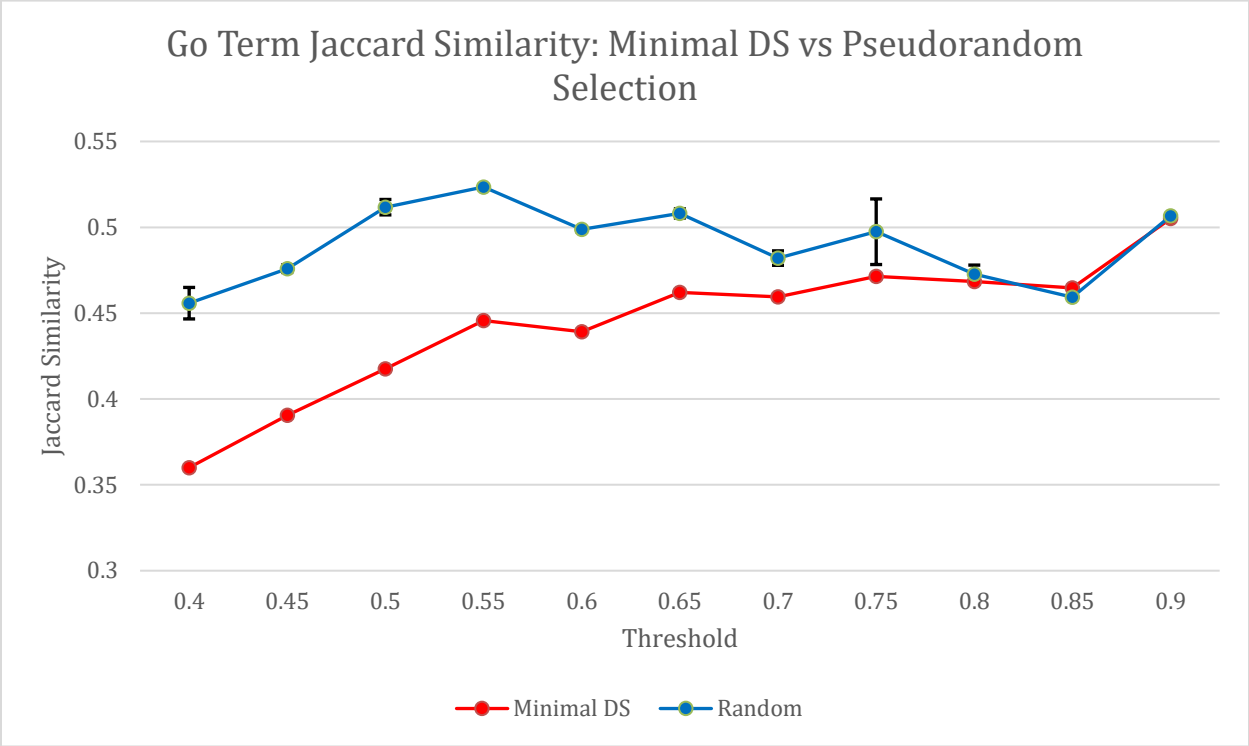


Figure 6: *The effect of minimality relaxation. Comparisons were repeated as described in Figure 5, but with minimal dominating sets. While relatively fast, minimal approximations to MDS failed to outperform pseudorandom selection at any threshold tested.*

Second, there was minimal enrichment for GO terms, suggesting a diverse sampling of a broad range of biological functions or unknown functions. To date, 70 of the MDS selected genes have entered into the production pipeline and 40 individual mouse strains have been established. The expansion and phenotyping of these novel mouse lines is currently underway. The information obtained from these tested mice will help to fill knowledge gaps and increase connectivity of our knowledge graph.

4.2 Study Limitations

Systems biology methods are currently limited by the accuracy of the chosen input data sources used to construct an initial knowledge graph. The source and construction of this graph will influence the outcome, and it is essential that this graph reflect the experimental application. Incorporating additional sources of information such as tissue specific co-expression data may help to provide context-dependent information critical for elucidating causal relationships in complex disease. For example, a gene-similarity graph containing neural co-expression data may increase the predictive power for identifying genes involved in behavior. Further, validation of different computational methods on biological networks are limited by what researchers in the past have prioritized. Such biases may have produced a more homogenous subset of genes when compared to the entirety of the gene-similarity graph. Thus, a small number of genes could potentially represent a large proportion of the GO terms associated with the set.

4.3 Directions for Future Research

The use of other graph coverage algorithms, such as independent set and vertex cover, warrants investigation for use in experiment selection. These other algorithms may have properties that researchers find desirable when selecting experiments to perform. For instance, given pairwise similarity between biological entities, researchers may want to select a subset in which no two entities are

similar, which independent set guarantees by a set of vertices with disjoint closed neighborhoods. Vertex cover, on the other hand, can be used if the coverage of all pairwise associations is desired. It should be noted, however, that the size of an MDS will always be equal to or smaller than the previously mentioned coverage algorithms. For this reason, we believe it is a reasonable approach when trying to maximize limited experimental resources.

We note that large swaths of the gene-similarity graph may be represented by a single gene in the MDS, for example the five clusters in Figure 4.A. The MDS is quite small compared to the size of the graph and may therefore comprise far fewer entities than resources allow for testing. In such a case, our method may be expanded with additional procedures that can be applied to expand the set to a desired size. For instance, one could recursively select a neighborhood covered by a member of an MDS, for example Cluster 1 in Figure 4.A, and then determine an MDS of Cluster 1 at a more stringent threshold and repeat until sufficient domination is achieved. Such a procedure, however, would need to be utilized under considerations of resources available for experimental validation.

Finally, our method is highly amenable to other systems biology contexts where resource, time, or methodological constraints do not allow comprehensive testing of all elements in a system. Other applications include reducing the scope of a compound library screening, and prioritizing members of a microbiome to perturb its metabolic networks. Each different application will also be sensitive to input data and thus, it is critical to investigate further how different input data sources influence network topology.

Through the application of a graph-theoretical algorithmic approach toward maximization of knowledge graph domination, large-scale research programs such as the IMPC can make the most advantageous use of limited resources to make the first inroads into characterization of poorly studied genes.

CHAPTER IV

THE SIGNIFICANCE OF PREPROCESSING IN THE CONTEXT OF POPULATION-BASED DATA ANALYSIS

This chapter is from a manuscript in preparation of the same title authored by Stephen K. Grady, Paul D Juarez, and Michael A. Langston. My contributions include method development, implementation, and testing.

ABSTRACT

In recent years, the exposome has become a popular framework in which to study environmental exposures from conception to death and the role these exposures can play in human health. Exposome data have therefore become an increasing important resource in the quest to untangle complicated health trajectories and help connect the dots from exposures to outcomes. This approach can be plagued, however, by noise in such forms as missing, duplicated, conflicting, incompatible, incorrect, and/or outlying data that can stymie downstream combinatorial and statistical analyses. Another problem common to this approach is multicollinearity, which frequently arises from repeated measurements taken over time and from multiple sources. Here the significance of preprocessing is described in the context of exposome analytics. Major concerns and strategies for dealing with noise are described. A novel graph theoretical technique for reducing the effects of multicollinearity is also introduced and analyzed. Empirical results using the Public Health Exposome are reported.

1. INTRODUCTION

The effects of environmental factors on human health have received considerable attention in recent years. The Exposome is a framework through which the modulating effects of the environment on human health can be studied [81]. To this end, several databases are now used to collect and curate exposure measurements. Notable examples include the Public Health Exposome (PHE) [82], the Southern Community Cohort Study (SCCS) [83], the Toxic Exposome (T3DB) [84], the Comparative Toxicogenomics (CTD) database [85], the Human

Early Life Exposome Study (HELIX) [86], Exposome Explorer [87], and the Geoscience and Health Consortium (GECCO) [88].

These databases generally incorporate information from a multitude of heterogenous sources. This results in high dimensional structures that can lead to numerous problems in downstream analysis. Noise, for example, may present itself in a variety of forms. These include missing, duplicated, incorrect, inconsistent, or outlying data that may arise from faulty sensors, incomplete and self-reported surveys, or legally required data suppression. In addition, noise can be found in the varying standards and practices of the sources from which data is collected that can lead to conflicting measurements. Take for example, two data sources: one that records a zero when no measurement is taken, and the other that records a zero when it is truly measured. While their values are the same, their meanings are quite distinct. Heterogenous sources may also introduce confusion due to incompatible or mixed data types, such as the recording of numerical and categorical data.

Multicollinearity is another thorny problem and one common to public health data. It can sometimes be found in the form of autocorrelates arising from subsets of variables that contain information for an exposure that is repeatedly measured over a given timeframe, say year-to-year, with little to no change. It can also be found in similar measurements taken from differing sources, for example, data for ambient air temperature taken from both the Environmental Protection Agency and the National Aeronautics and Space Administration. Negative impacts of multicollinearity can be seen in the statistical analyses employed in exposome studies [89, 90], particularly in regression analysis and the many methods built upon it [91, 92].

Given these issues, we present a series of preprocessing methods that can help edit, clean, standardize, and harmonize public health data. In Section 2, we

describe in some detail the PHE, its data, and its data dictionary. In Section 3, we discuss a set of preprocessing tools including those used to address noisy and messy data. In Section 4, we present supervised and unsupervised feature selection techniques. In Section 5, we introduce a novel graph theoretical technique to reduce the effects of multicollinearity through reductions in autocorrelates. In Section 6, we report empirical results of these methods, focusing on metrics such as their ability to reduce skew in correlation distributions and to improve resultant cluster quality. In a final section, we close with concluding remarks, study limitations, and possible avenues for future research.

2. *The Public Health Exposome*

The PHE database was created and is maintained at Meharry Medical College as a central repository for storing environmental measurements from diverse sources such as the Centers for Disease Control Wonder database [93], Dartmouth Atlas of Healthcare [94], the United States Census [95], and the Environmental Protection Agency [93], to name but a few. To date, the PHE is comprised of over 52,000 variables recorded at every one of the 3,141 counties, parishes, and boroughs in the United States. Its variables can be classified into five environmental domains: built, health care, natural, policy, and social environment. The PHE has seen use in a variety of studies, such as determining the role the environment plays in the development of cardiovascular disease [96], obesity [97], lung cancer [98], preterm births [99], and health disparities [100]. The PHE contains a data dictionary that holds meta information for each variable. This information includes the domain and category to which each variable belongs, the year in which each variable was recorded, and information pertaining to race and sex specifics when applicable. Importantly, the dictionary contains descriptions for each variable that can be used to distinguish what is being measured. The PHE is an apt exemplar for the present study largely because it contains an enormous and heterogenous set of data that exhibits all

the problems described in Section 1. We are primarily interested in methodological comparisons. But computational recalcitrance limits duration. We therefore concentrated our efforts on variables recorded since 2014. This subset of 6,694 variables comprises the largest, most diverse, and hopefully most representative collection of PHE database entries.

3. *Noise Reduction and Data Cleaning*

Even the most well-designed studies will run into issues posed by data irregularities. Data cleaning [101], the detection and correction of data abnormalities, is an integral first step in any preprocessing toolchain. Methods employed for this purpose are highly dependent on study contexts, objectives, and sources of data errors. If domain expertise is available, outlying data can be addressed by determining a realistic range for measurements outside which they may be discarded. If outliers cannot be discarded, data transformation via normalization techniques may be utilized to deemphasize their effects. Variables with missing measurements may be discarded if they are not crucial to a study. A threshold of more than 40% missing values has been found to be a good cutoff [102]. If, on the other hand, variables are known to be of importance to a study, imputations methods [103] may be deployed. It is important to note, however, that variables with missing data may be suppressed due to small case size resulting in unreliable rates, or due to confidentiality concerns, which must be taken into consideration.

Variance should be considered as well. A variable without variance provides no information and should of course be eliminated. Practitioners may also wish to discard a variable with variance below some nonzero threshold, but there seems to be no universal standard for that.

Non-numerical variables such as categorical data should be transformed to numeric values before any downstream analyses may take place. Researchers

may choose to convert such variables to binary variables through dummy coding or to ordinal variables through integer ranking systems before being used in similarity metric computations. Incompatible data requires domain expertise to rectify as each source of data may have its own standards for how to mark a missing value vs a true measurement.

4. *Feature Selection Methods*

Exposome studies often begin with immense volumes of high dimensional data from which only a small proportion will prove useful. Numerous variables may be irrelevant or redundant. Others may prove useless in a given application. Feature selection is a process that can help reduce the number of variables with which one is working to home in on those of interest and improve the interpretability of downstream analyses. Feature selection methods can be divided into two categories, namely, supervised and unsupervised.

Supervised methods are frequently employed when one is studying a specific variable, for instance a health outcome, and would like to find a subset of variables that are related, or influence said variable. A simple measure of similarity may suffice to select all features that are above a given threshold. Some examples include Pearson's correlation coefficient [104] and mutual information [105]. Each comes with its own drawback, however, as mutual information may not always be the best choice when regression analysis is to be used with the selected features [106], and Pearson's correlation coefficient may not pick up on nonlinear similarities.

More advanced supervised methods to capture non-linear associations often depend on a machine learning model. Machine learning methods that utilize a tree-based structure such as Random Forest [107], Iterative Random Forest [108] and Gradient Boosted Trees [109] are natural candidates to be used for feature selection tasks due to their ability to score each variable used in its

model. The feature importance score may be determined using Gini importance [110], permutation based accuracy measures [111], or Shapley values [112]. These models and the feature importance scores they provide have been used in exposome studies to determine obesity rates [113]. When using feature importance for the purpose of feature selection, a suitable threshold above which a feature is retained must be determined. Such a threshold may not always be apparent. Recursive feature elimination [114] addresses this problem by recursively training a model and removing features with the lowest importance scores until either a desired number of features are selected or all subsets have been tested. In the latter case, the subset that was used to train the best performing model is retained. It should be noted that this method can fail to scale when used with a large number of variables [115]. For a full review on supervised feature selection see [116].

Unsupervised methods, on the other hand, can be used when one is searching for latent relationships among variables of interest. Generally, unsupervised feature selection methods are used before a task such as clustering [117], and often in exposome studies, latent networks (clusters) are desired [98, 99, 118]. In these situations, unsupervised feature selection methods are appropriate. For a review on unsupervised feature selection methods see [119]. Graph theoretical algorithms, with their ability to scan the entire solution space, are particularity well suited to such a task. They have been utilized for unsupervised feature selection using methods such as clustering [120, 121], centrality measures [122], a hybrid of the two [123, 124], vertex cover[125], and spectral methods [126]. Each of these methods models data as a graph from which a subset of vertices is then selected such that variable redundancy is reduced.

5. *A Graph Theoretic Approach to Autocorrelation Reduction*

In this section we present the use of a graph theoretic approach to reduce autocorrelation in public health data. We first introduce some graph theoretic

terms and preliminaries. The neighborhood of a vertex *u* consists of *u* and all vertices adjacent to *u*. Recall that a minimum dominating set is a dominating set of smallest size, and a minimal dominating set is one that cannot be made smaller but is not necessarily one of smallest size. A vertex *u* dominates a vertex *v* if *u* and *v* are adjacent and *u* but not *v* is in a dominating set.

We now introduce a novel unsupervised feature selection method to address the problem of multicollinearity, by reducing the prevalence of autocorrelates. We first constructed a graph where vertices represented PHE variables and an edge was placed between any two vertices if and only if a two-fold threshold was passed. Edges were weighted by both measure similarity from variables' recorded data and semantic similarity from their corresponding variable descriptions. For the first threshold, we computed Pearson's correlation coefficient between all pairs of variables. We adapted the guideline that any pair of variables with a Pearson's correlation coefficient greater than or equal to 0.90 can be considered autocorrelates [127]. For the second threshold, we constructed extended variable descriptions for each variable using all available metadata. Using all descriptions as a corpus, we used the tf–idf (term frequency inverse document frequency) [128] algorithm provided by the open source machine learning python library *scikit-learn* [129] to compute representative vectors for each description. Vectors were compared using cosine similarity [130]. Finally, an edge was placed between a pair of vertices if an only if their Pearson's correlation coefficient was greater than 0.90 and their cosine similarity was greater than or equal to 0.20. There are no guidelines for the latter threshold so a value of 0.20 was chosen as this is where an inflection point was identified with an elbow plot of cosine similarities (Figure 7). We refer to the resultant object as an autocorrelates graph.

We sought a subset of vertices that would represent all variables found in the autocorrelates graph while reducing their number. For this purpose, we applied

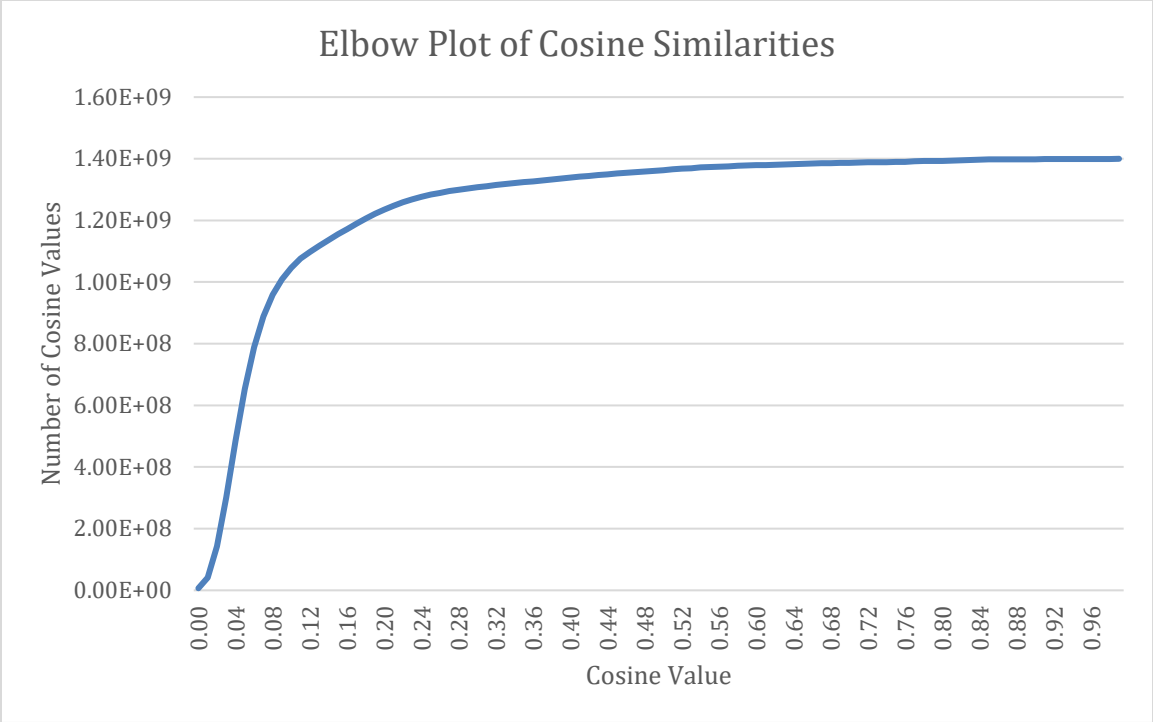


Figure 7: *An elbow plot of cosine similarities generated from PHE metadata. The inflection point at 0.20 was selected as a threshold for cosine similarities.*

a minimum dominating set solver to select a set of variables, and then removed all variables not in the selected set. To solve minimum dominating set we used the open source mathematical optimization solver Cbc (Coin-or branch and cut) [131] through the Python package *Python-MIP* [132]. We chose this open-source option over a commercial solver as an academic license was not available at the facility where computations took place. We used the standard ILP formulation

$$\textit{minimize} \sum x_i \; \forall \; i \in V \textit{ subject to } \sum x_i \geq 1 \; \forall \; i \in \mathrm{N}[v] \text{ where } x \in \{0,1\}$$

where x is an integer variable corresponding to a vertex $v \in V$. A complete flowchart of our preprocessing method can be found in Figure 8.

In certain problem settings solving minimum dominating set may be infeasible as it both *NP*-complete [79] and W[2]-complete[4]. In such cases the use of a minimal dominating set may be more suitable. To determine a minimal dominating set we utilized a greedy algorithm [80] that takes only time linear in |E|. This method works by iteratively selecting a vertex with the most undominated vertices in its neighborhood to be in the minimal dominating set until no such vertex can be selected. Minimal dominating set was computed using custom code written in Python that utilized the *Networkx* [133] Python package. All computations were performed on the Cori high performance computing cluster maintained by the National Energy Research Scientific Computing Center (NERSC) [134]. Computations were completed on one of Cori's shared large memory nodes containing two AMD EPYC 7302 (Rome) 3.0 GHz processors, 42GB DDR4 memory and Cori's CSCRATCH read/write system.

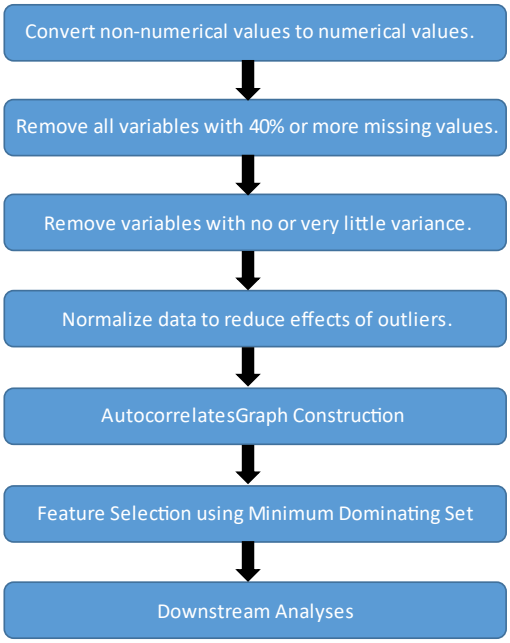


Figure 8: *A flow diagram of our preprocessing steps. First, noise reduction techniques are applied. All non-numerical values are converted to numeric values through dummy coding. Then all variables with 40% or more missing values are removed. Variables with little to no variance, depending on study priorities are then removed. For the final noise reduction step, all remaining variables are normalized to reduce the effects of outliers. After these steps, using the remaining variables, an autocorrelates graph is constructed. Finally, a subset of variables is selected using minimum dominating set to be used in downstream analyses.*

6. Empirical Evidence and Analysis

In this section we present results of applying preprocessing methods to address the problem of autocorrelation found in the PHE. We tested a noise reduction toolchain, the use of minimum dominating set, the use of minimal dominating set, and two previously used centrality measures. These methods, except the noise reduction techniques toolchain, were applied to an autocorrelates graph constructed as described in Section 5.

Each method produced a subset of variables from which we determined a Pearson's correlation coefficient distribution, the percent of correlation values 0.90 and above, and the distributions kurtosis. Correlation distributions were used to determine the presence heavy tails [135], that is the greater than expected presence of large values, a sign of autocorrelation. Values 0.90 and greater were a direct measure of the presence of autocorrelation as previously defined. Excess kurtosis was used to measure the peakedness of the distributions, where a value of zero represents a normal distribution and larger positive/negative values denote larger/lower peak. Kurtosis was computed using the statistics library found in the Python package *Scipy [136]*. In addition, for each of subset of selected variables we constructed a graph where vertices represented PHE variables and edges were weighted by Pearson's correlation coefficient. We then computed an appropriate threshold using spectral methods [23] to construct an unweighted graph. To this graph we applied the paraclique algorithm [137] to decompose it into a set of clusters. For each paraclique (cluster) we determined the average cosine similarity between all members' corresponding variable descriptions. We used these averages to determine a paraclique's diversity which we define as the inverse of its average cosine similarity. Therefore, higher paraclique diversity is desired as this means tightly correlated variables with little descriptive similarity are being clustered together, instead of autocorrelates.

6.1 The Base PHE

For comparison purposes, we first determined the metrics just described for the PHE before any preprocessing steps were applied. The distribution of correlation values had a heavy right tail (Figure 9) suggesting the presence of autocorrelation. Kurtosis was determined to be 0.67. Roughly 1.3% of the correlation values were 0.90 or higher. The average paraclique cosine similarity tends to be higher near 0.90 indicating a lack of diversity in paraclique membership and a presence of autocorrelation (Figure 10).

6.2 Noise Reduction Techniques Results

We first tested a toolchain of noise reduction techniques. The toolchain consisted of correcting incompatible data, removing variables with missing values and no variance, and data normalization. After applying this toolchain, we again generated a correlation distribution and paraclique diversity metrics. After applying these steps, the PHE was left with 4,704 variables. The noise reduction steps did little to change the heavy right tail of the PHE's correlation distribution (Figure 11). Kurtosis was reduced to the lowest of any method tested at 0.02. The percent of correlation values equal to 0.90 or greater was 1.8%, an increase over that of the base PHE. Upon further investigation, we discovered that of the correlation values between variables removed by the noise reduction steps only 0.4% were 0.90 or greater. Since the grand majority of correlation values between variables removed were below this threshold, proportionally more autocorrelation remained. Paraclique member diversity also did not improve with a large proportion of average cosine similarities found around 0.90 (Figure 12). Taken together, these results demonstrate these techniques lack of ability to address autocorrelation, however, they do address kurtosis.

6.3 Minimum Dominating Set Results

As just presented, the use of standard noise reduction techniques made worse the heavy right tail in correlation distributions and the percent of correlation

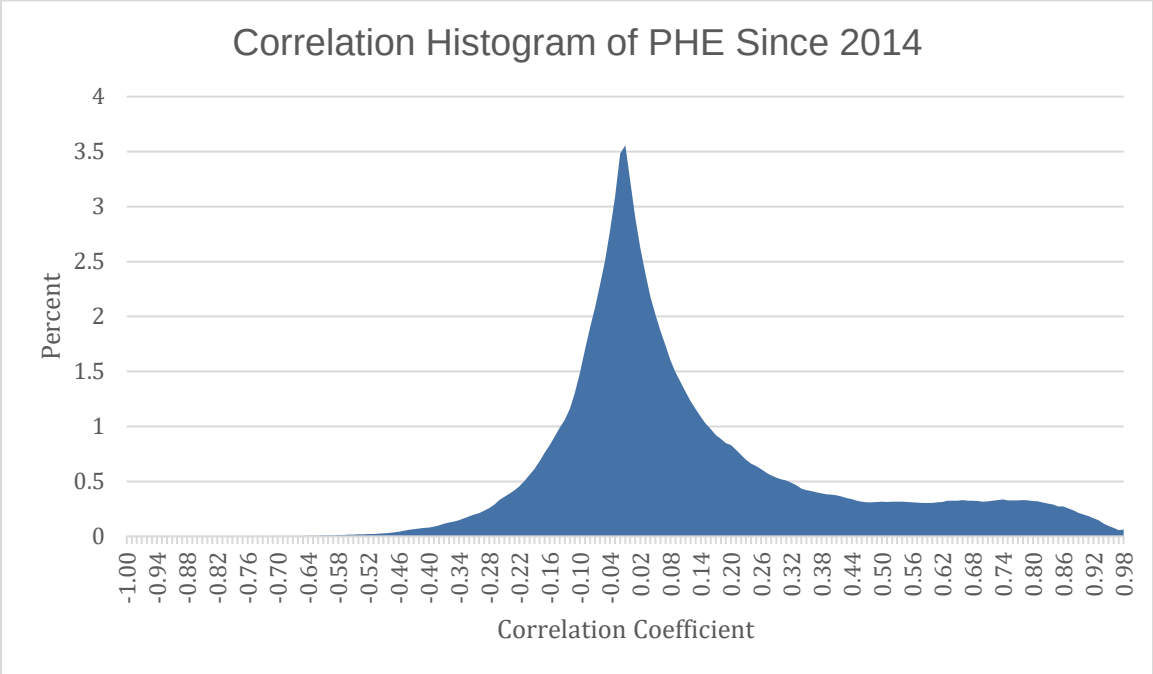


Figure 9: *The distribution of Pearson's correlation coefficients between all variables before preprocessing methods were applied. Note the heavy right tail, indicative of autocorrelation.*

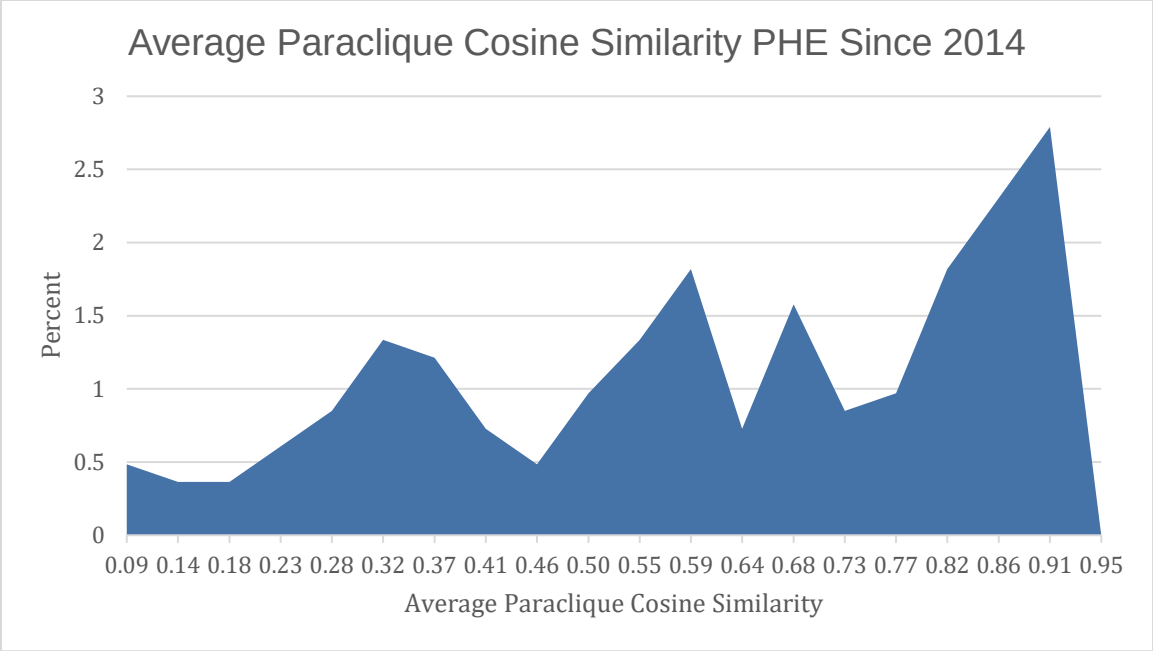


Figure 10: The distribution of the average cosine similarity between paraclique members for all paracliques without any preprocessing methods applied. Note that the average paraclique similarities tend towards 0.90 indicating paracliques that contain variables with highly similar variable descriptions indicating low variable diversity and the presence of autocorrelates.

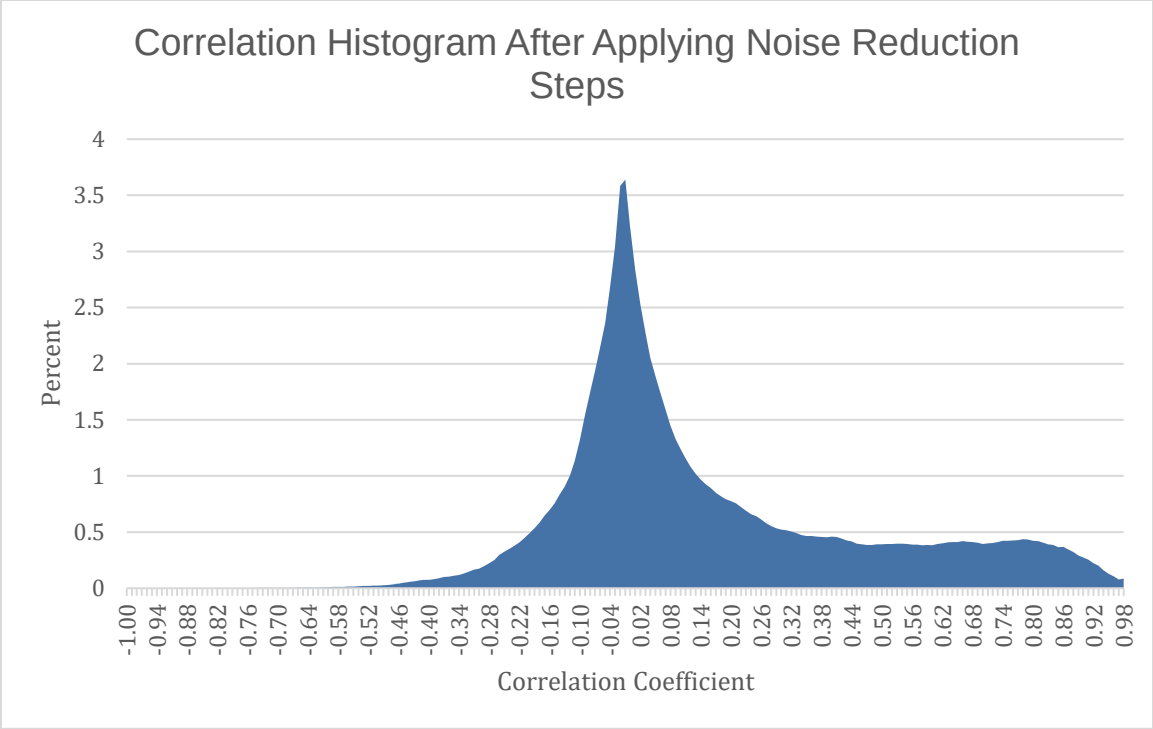


Figure 11: *The distribution of Pearson's correlation coefficients after applying the noise reduction steps. The presence of a heavy right tail persisted and was made even worse above 0.90 when compared to the base PHE.*

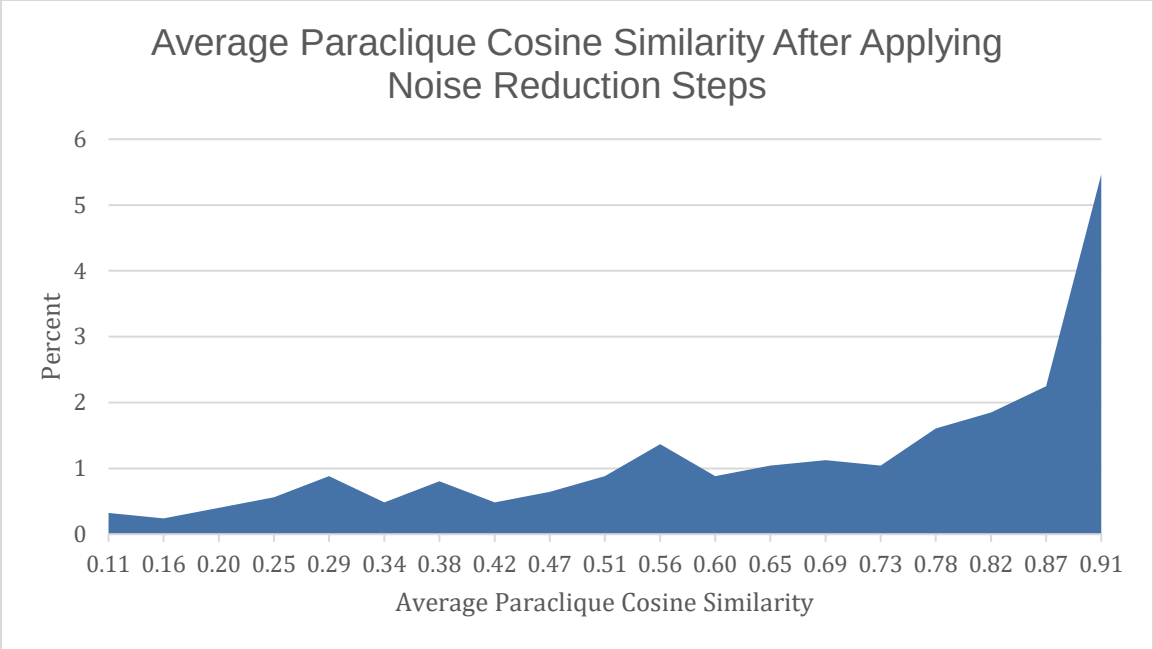


Figure 12: *The distribution of the average cosine similarity between the paraclique members for all paracliques from the PHE after applying noise reduction steps. Note that most averages tend to be larger than 0.75 indicating a lack of diversity in paraclique membership suggesting the presence of autocorrelates.*

values 0.90 or greater. They also did little to improve paraclique membership diversity. Turning to the use of minimum dominating set we present two sets of results. For greater insight into the utility of the two-fold threshold method in Section 5, we also tested the effects of applying minimum dominating set to an autocorrelates graph constructed using only the Pearson's threshold. When minimum dominating set was applied to the Pearson's only graph, it reduced the PHE to 1,638 variables. Its correlation distribution was greatly improved when compared to the use noise reduction techniques (Figure 13). The percent of correlation values 0.90 or greater was vastly reduced to 0.008%.

Kurtosis (2.289) was increased over that of the base PHE and noise reduction techniques, and in fact was the largest for any method tested. Its paraclique diversity profile, however, was not improved over that of noise reduction techniques (Figure 14). This may have been due to the relatively fewer paraclique produced from its resulting graph. Using the two-fold method, minimum dominating set reduced the PHE to 1,671 variables. Its correlation distribution was also greatly improved when compared to that of noise reduction techniques (Figure 15). The percent of correlation values greater than or equal to 0.90 was 0.03%. Kurtosis was once again high at 2.25. The paraclique diversity profile was greatly improved with the average cosine similarity skewing towards 0.25 suggesting a lack of autocorrelates in clusters (Figure 16). Minimum dominating set performed well using both graphs, however, the lack of paraclique diversity when only using Pearson's correlation coefficient to determine autocorrelates lends evidence to the utility of the two-fold threshold method. Kurtosis was highest when minimum dominating set on either graph. This is to be expected as these methods perform well at removing correlation values in the right tail, thus, leaving a greater proportion of values around the mean increasing either distributions' peak.

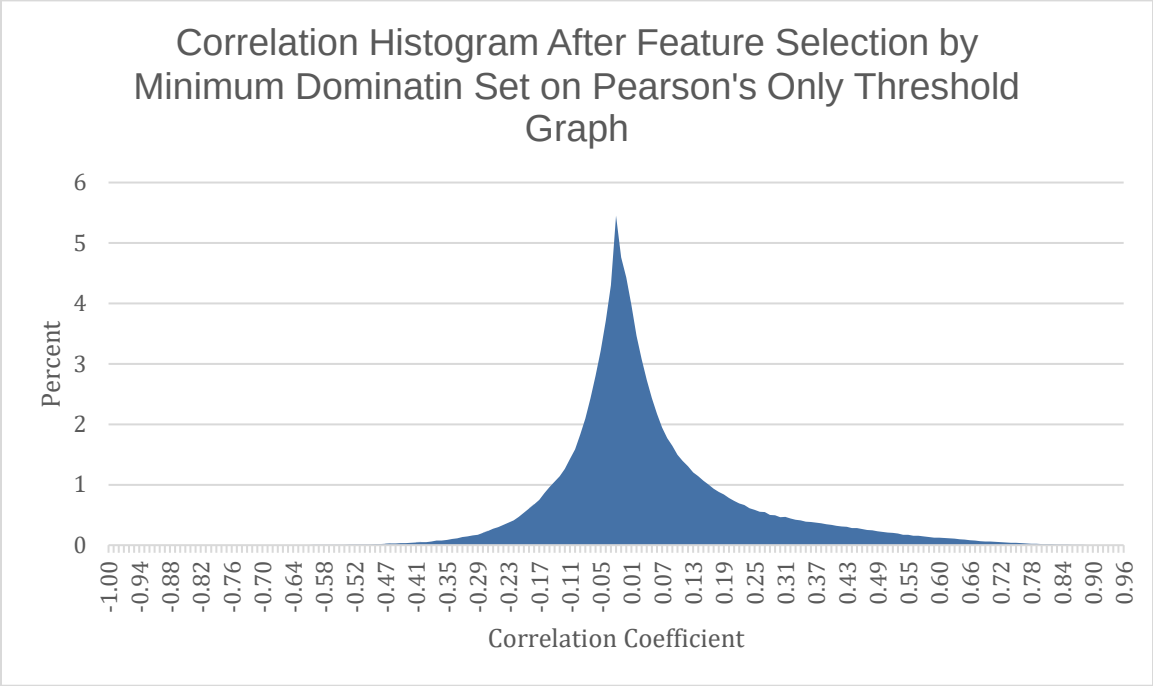


Figure 13: *The distribution of Pearson's correlation coefficients after variables were selected by minimum dominating set to an autocorrelates graph constructed using only Pearson's correlation coefficient as a threshold. The heavy right tail found in correlation distributions for the base PHE and the subset of variables after noise reduction techniques have been applied has been greatly reduced.*

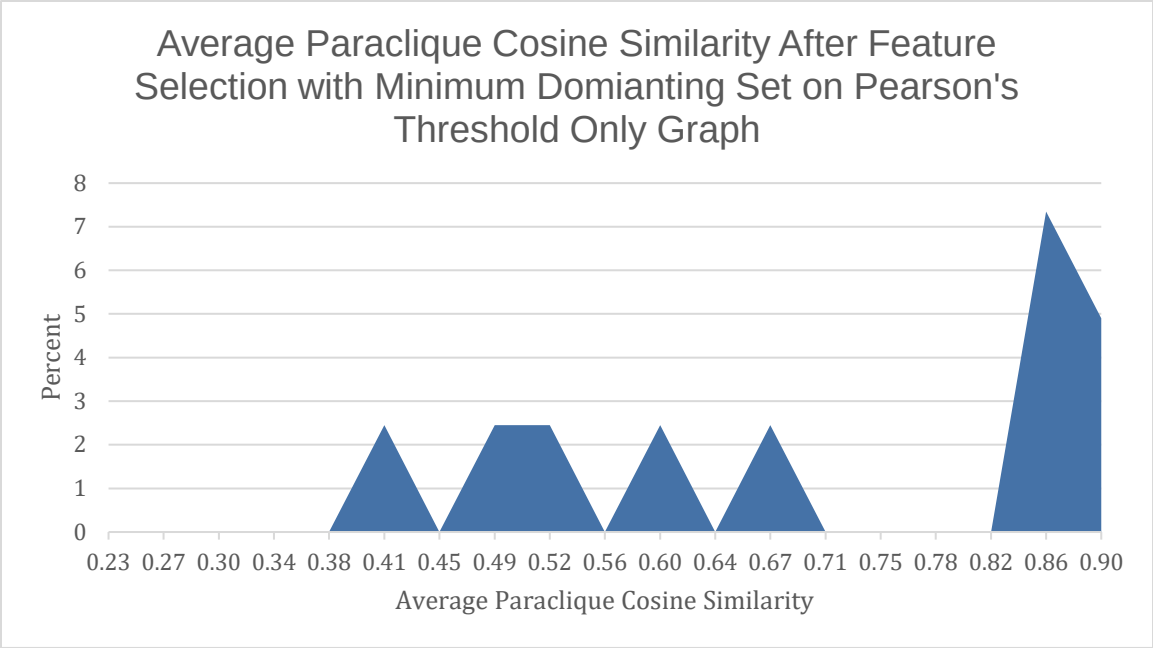


Figure 14: *The distribution of the average cosine similarity between paraclique members for all paracliques after variables were selected by minimum dominating set. The autocorrelates graph used was constructed with a Pearson's correlation coefficient threshold only. Paraclique diversity was not improved, however, this may be due to the relatively few paracliques extracted from the graph derived from the selected variables.*

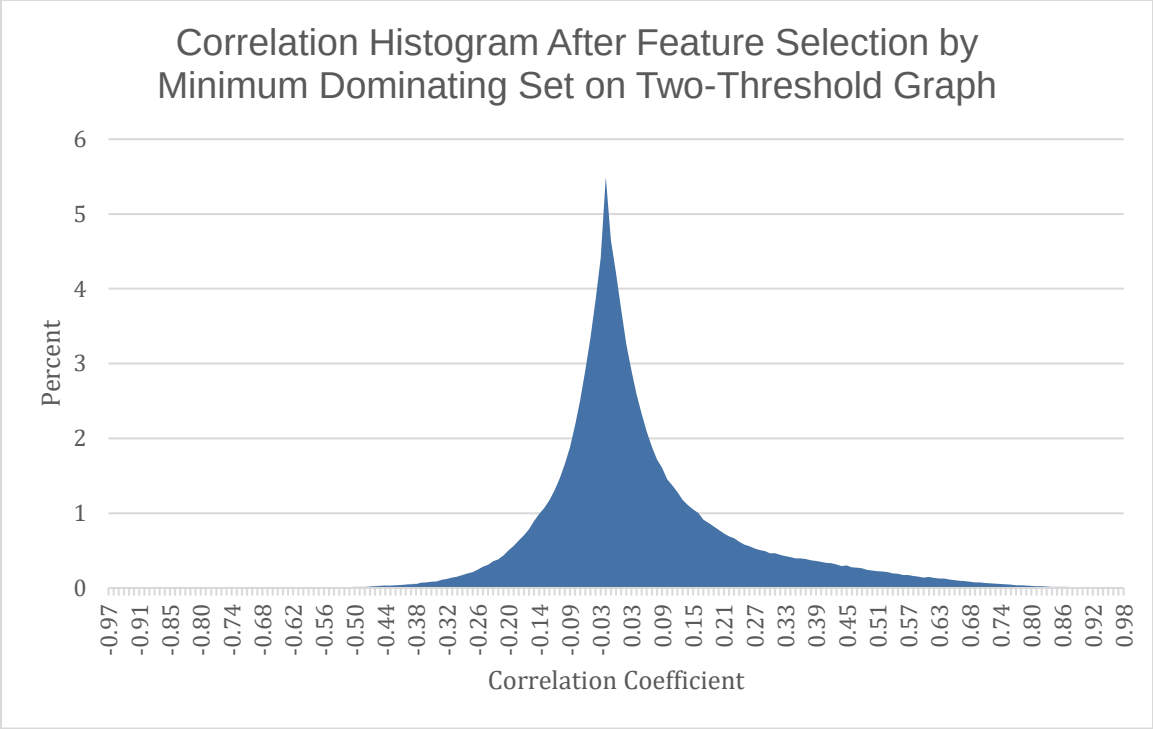


Figure 15: *The distribution of Pearson's correlation coefficients after variables selected by minimum dominating set applied to an autocorrelates graph constructed using the two-fold threshold method. Note that the heavy right tail present in the correlation distributions for the base PHE and the subset of variables after applying noise reduction techniques is greatly improved.*

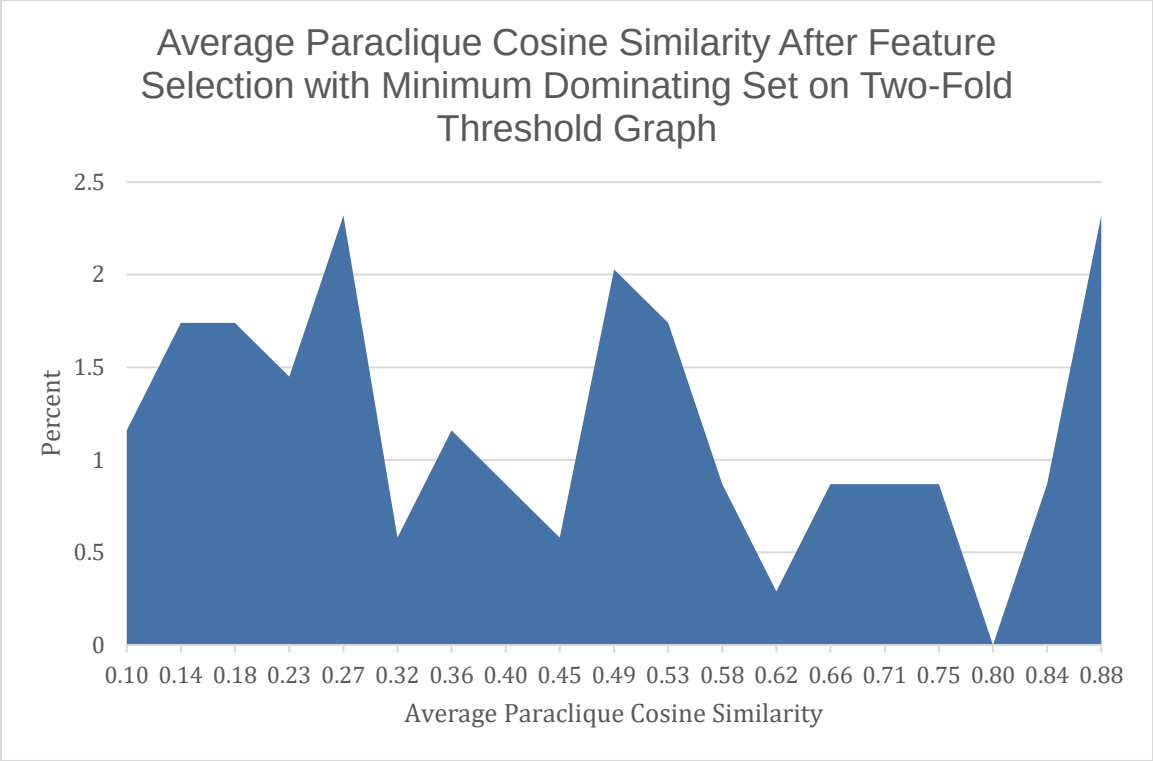


Figure 16: *The distribution of the average cosine similarity between paraclique members for all paracliques after variables were removed from the PHE by minimum dominating set applied to an autocorrelates graph constructed using the two-fold threshold method. The average paraclique cosine similarity tends to be lower suggesting paracliques with greater membership diversity and a reduction in autocorrelates. Paraclique diversity is also improved when compared to Figure 14, lending evidence for the utility of using the two-threshold method.*

6.4 Minimal Dominating Set Results

We tested the ability of minimal dominating set to reduce autocorrelation in an attempt to determine if the extra computation needed to solve minimum dominating set was worth the effort. Minimal dominating set reduced the PHE to 1,673 variables and its correlation profile was better than that of the noise reduction techniques but does not perform to the level of minimum dominating set (Figure 17). Its percent of correlations values 0.90 or greater was 0.8% and its kurtosis was 0.51. Its paraclique diversity distribution performed better than that of the noise reduction techniques and is on par with minimum dominating set (Figure 18). While the heavy right tail in the correlation distribution after applying minimum dominating set, demonstrating the latter's utility even with the need for extra computation.

6.5 Comparison to Centrality Measures

Finally, we tested the ability of two centrality measures to handle autocorrelation. Centrality measures are often used to vertices that in some sense cover a large portion of a graph. This has made them attractive for feature selection in the past. We tested four well known metrics of centrality, betweenness centrality [138, 139], Page rank [122, 140], degree centrality [139, 141], and eigenvector centrality [142, 143] which has been previously used for feature selection. In the same vein as testing minimal dominating set, we did this to determine if their simpler computation could perform as well as minimum dominating set in reducing autocorrelates. Both betweenness centrality and Page rank return a weight for each vertex, so we selected the top *k* vertices where *k* equaled the number of vertices selected by minimum dominating set.

The Pearson's correlation histogram after using betweenness centrality still contained a heavy right tail especially at 0.90 and above (Figure 19) and performed worse than either minimum dominating set or minimal dominating set.

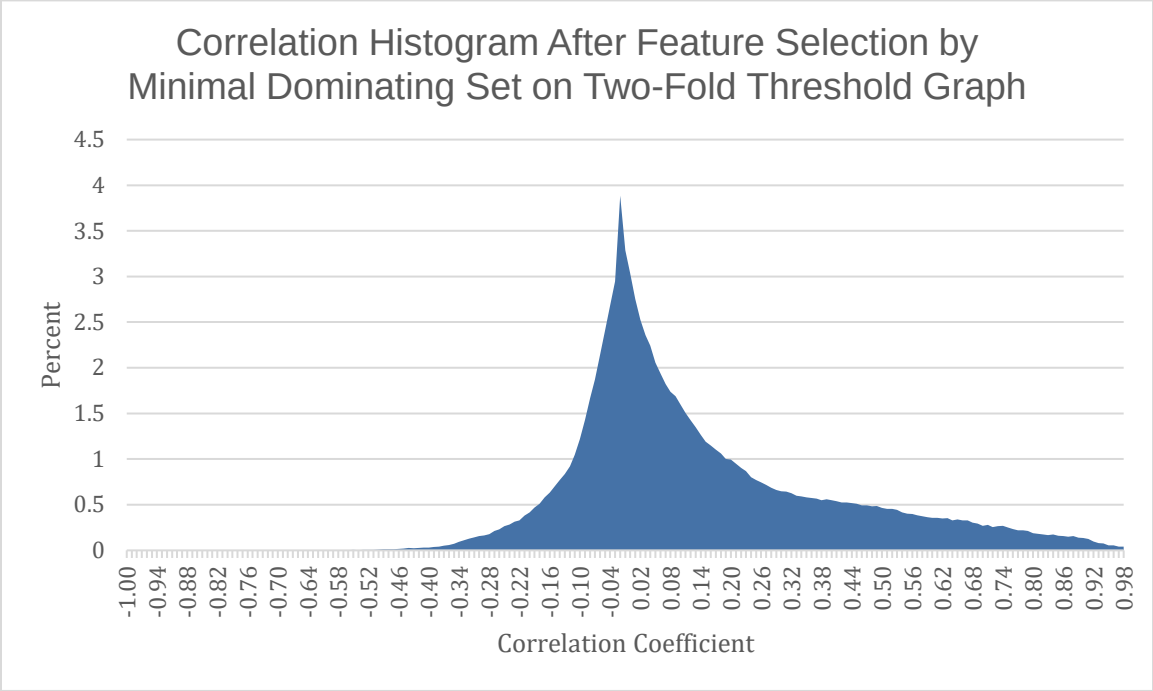


Figure 17: *The distribution of Pearson's correlation coefficients after variables were selected by minimal dominating set applied to an autocorrelates graph constructed using the two-fold threshold method. There is an apparent reduction in the heavy right tail, however it is not as pronounced when compared the reduction due to minimum dominating set.*

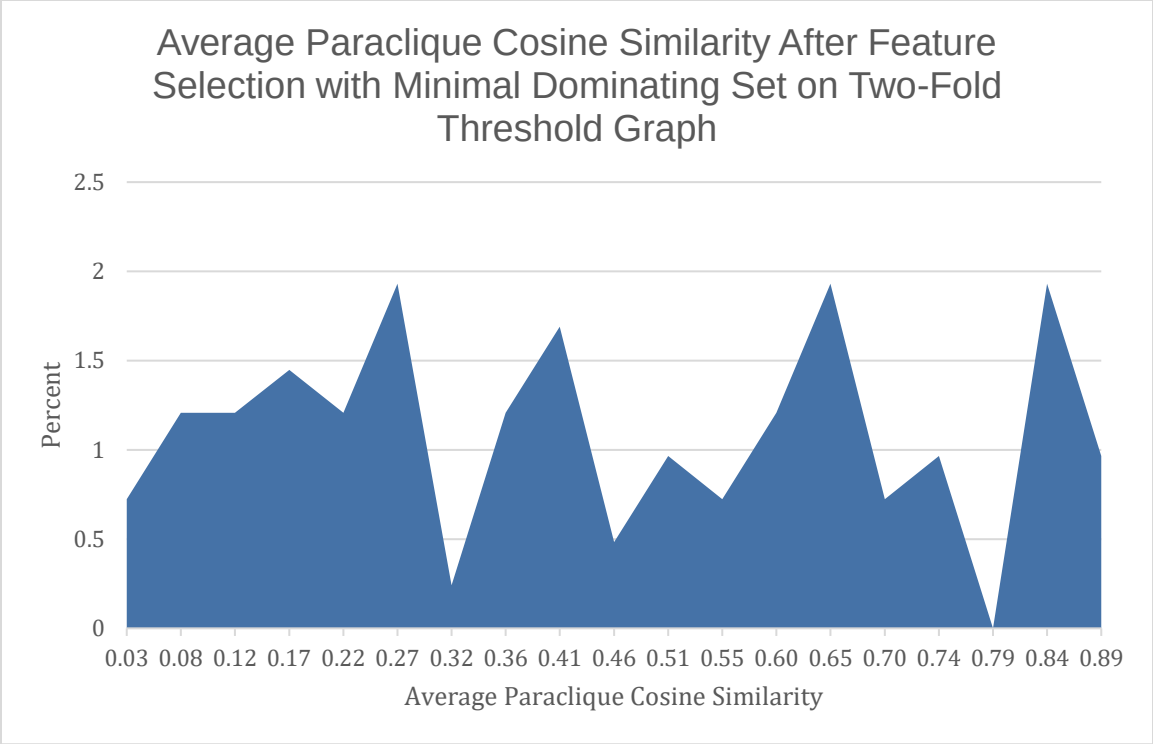


Figure 18: *The distribution of the average cosine similarity between paraclique members for all paracliques after variables were selected by minimal dominating set applied to an autocorrelates graph constructed using the two-fold threshold method. Paraclique diversity is improved as the average cosine similarity has a higher proportion towards 0.20.*

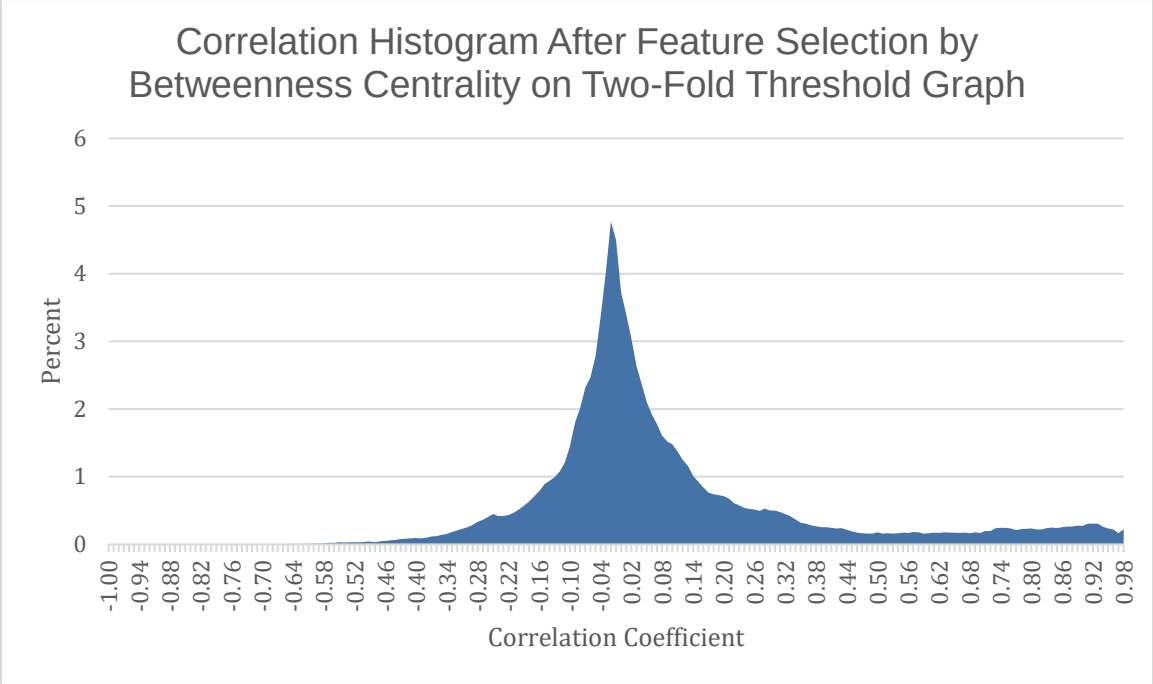


Figure 19: *The distribution of Pearson's correlation coefficients after variables were removed from the PHE by betweenness centrality applied to an autocorrelates graph constructed using the two-fold threshold. The heavy right tail has been reduced to an extent, but there still remains a large proportion, 2.5%, of values at 0.90 or greater.*

The percent of correlation values 0.90 or above was 2.5%. Excess Kurtosis was measured at 1.99. Its paraclique diversity distribution was an improvement over that of noise reduction techniques but was not on par with minimum dominating set or minimal dominating set (Figure 20). Page rank performed the worst with the heaviest right tail of all methods tested (Figure 21). Its percent of correlation values 0.90 and greater was 15.3%. It produced a negative kurtosis at -1.56, due to the large proportion of values in the right tail. The paraclique diversity distribution produced by Page rank (Figure 22) was an improvement over that of noise reduction techniques but did not outperform that of minimum dominating set or minimal dominating set. Degree centrality did not perform that much better, with the presence of the heavy right tail worsening (Figure 23) after its application and the proportion of correlation values 0.90 or greater increasing to 6.3%.

Excess kurtosis was negative at -1.12, again due to the large proportions in the right tail. Paraclique diversity (Figure 24) was improved over that of other centrality measures with a reduction in the proportion of average cosine similarity at 0.90 and an increase their proportion at values less than 0.55. Finally, eigenvector centrality performed much the same as degree centrality. Its heavy tail increased (Figure 25) with correlation values 0.90 or greater at 6.1%. Similar to degree centrality its excess kurtosis was -1.28 due to its heavy right tail. Eigenvector centrality did not perform as well as degree centrality in improving paraclique diversity (Figure 26), however, it was improvement over that the other centrality measures with a slight reduction in the proportion of average cosine similarities around 0.90. Compared to minimum dominating set, these four centrality measures failed to produce results in the reduction of autocorrelates comparable to that of minimum dominating set. In fact, all methods made autocorrelates more pronounced. These results, taken in light of those for minimum dominating set demonstrate its utility.

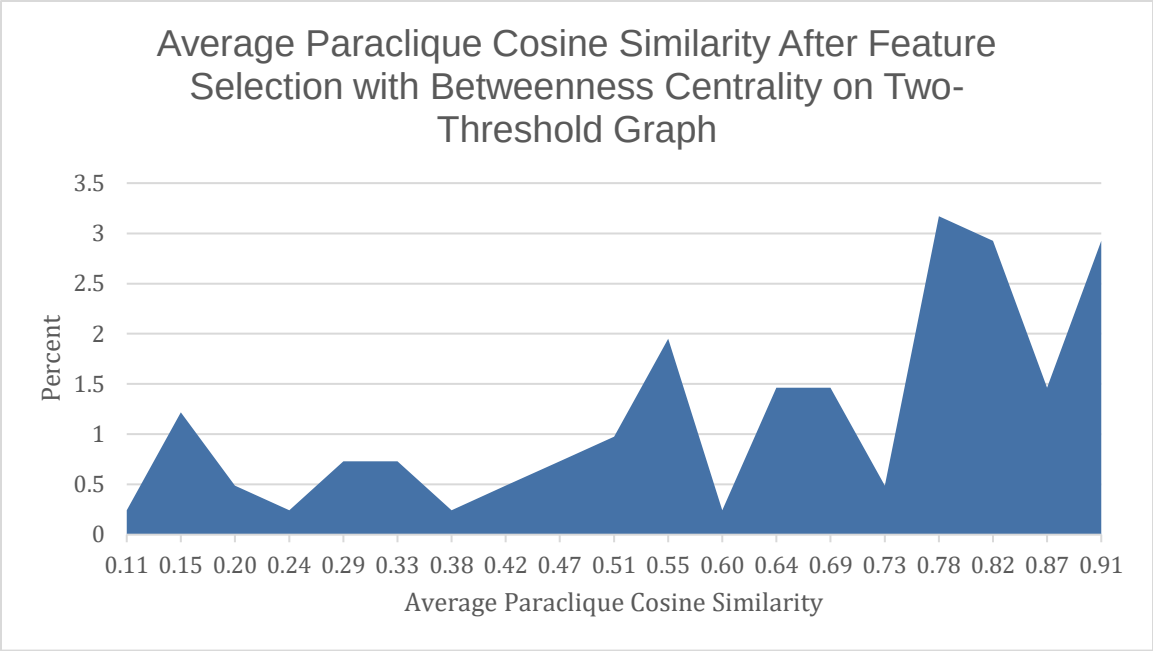


Figure 20: *The distribution of the average cosine similarity between the paraclique members for all paracliques from the PHE after variables were removed from the PHE by betweenness centrality applied to an autocorrelates graph constructed using the two-fold threshold. Paraclique diversity is only slightly improved as average cosine similarities tend towards 0.80.*

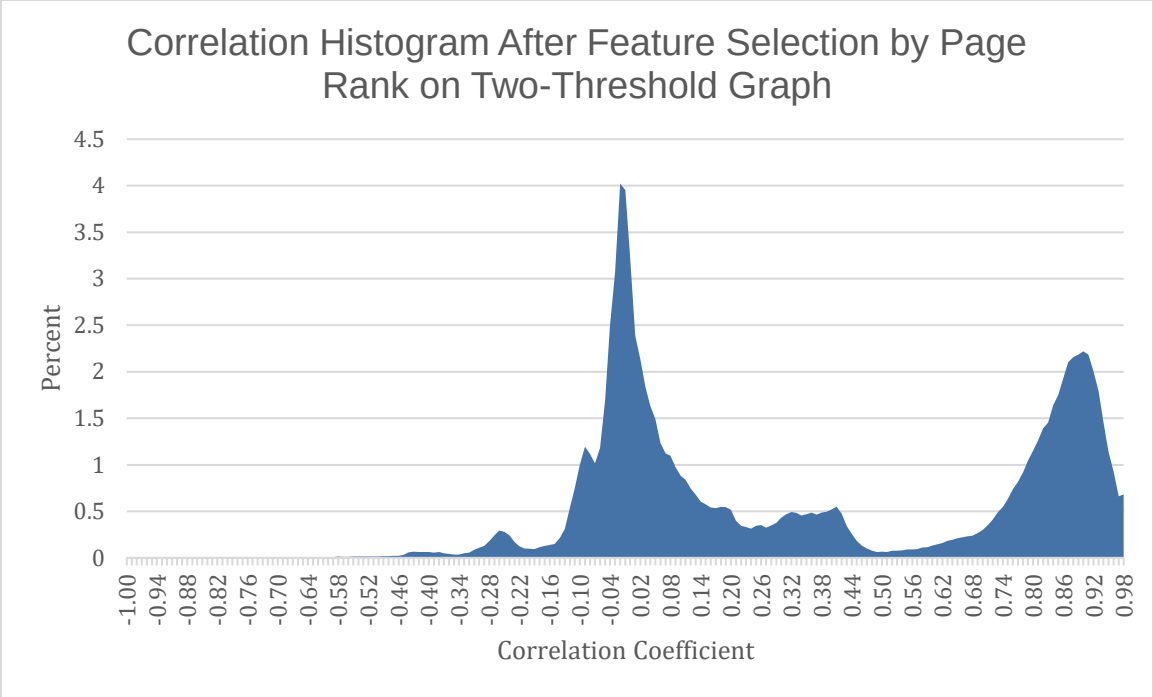


Figure 21: *The distribution of Pearson's correlation coefficients after variables were removed from the PHE by Page rank applied to an autocorrelates graph constructed using the two-fold threshold. The use of page rank performed the worse of any method tested. It removed variables that shared correlation values around 0.50, greatly increasing the proportion of values 0.90 and above to 15.3%.*

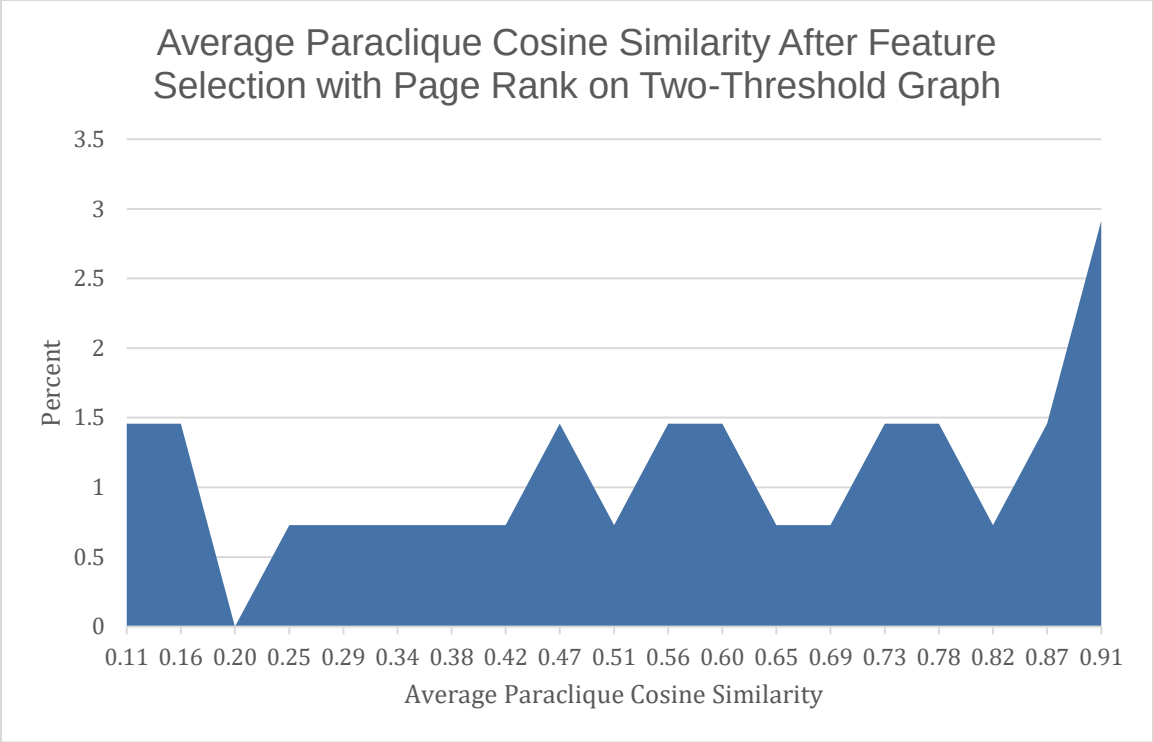


Figure 22: *The distribution of the average cosine similarity between the paraclique members for all paracliques from the PHE after variables were removed from the PHE by Page rank applied to an autocorrelates graph constructed using the two-fold threshold. Paraclique diversity did not improve much as to be expected with such large a large proportion of correlation values greater than or equal to 0.90.*

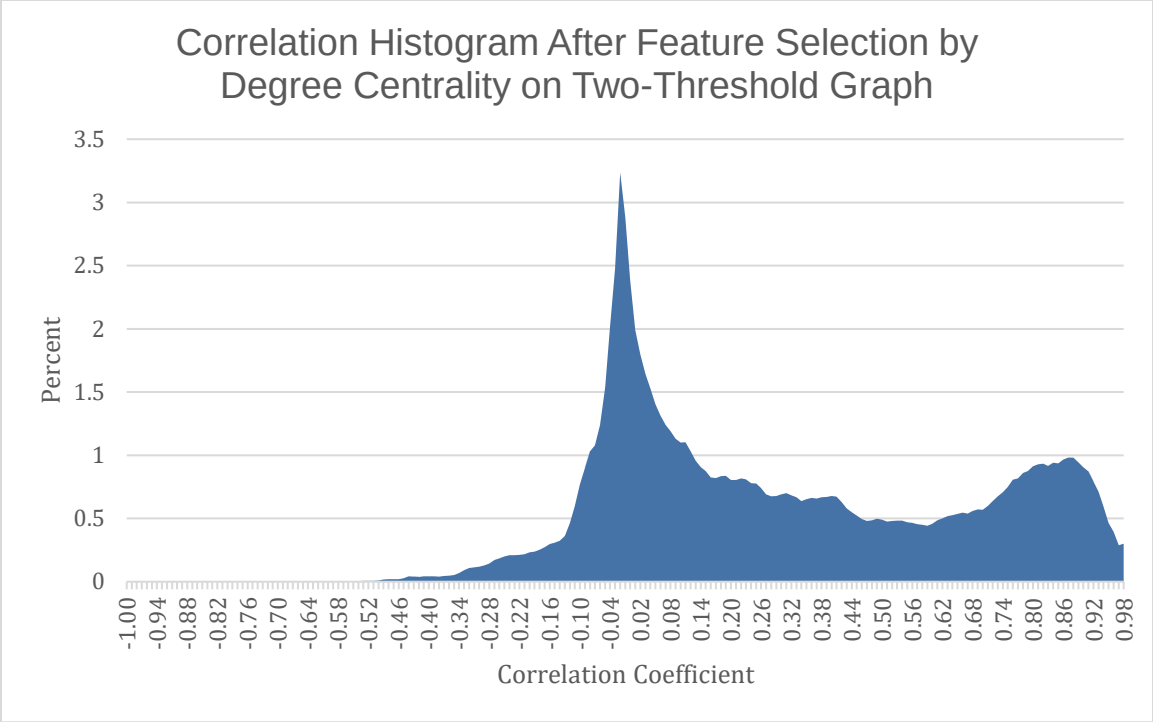


Figure 23: *The distribution of Pearson's correlation coefficients after variables were selected by Degree Centrality applied to an autocorrelates graph constructed using the two-fold threshold method. Degree centrality removed variables that shared correlation values around 0.50 as well, increasing the proportion of values 0.90 and above to 6.3%.*

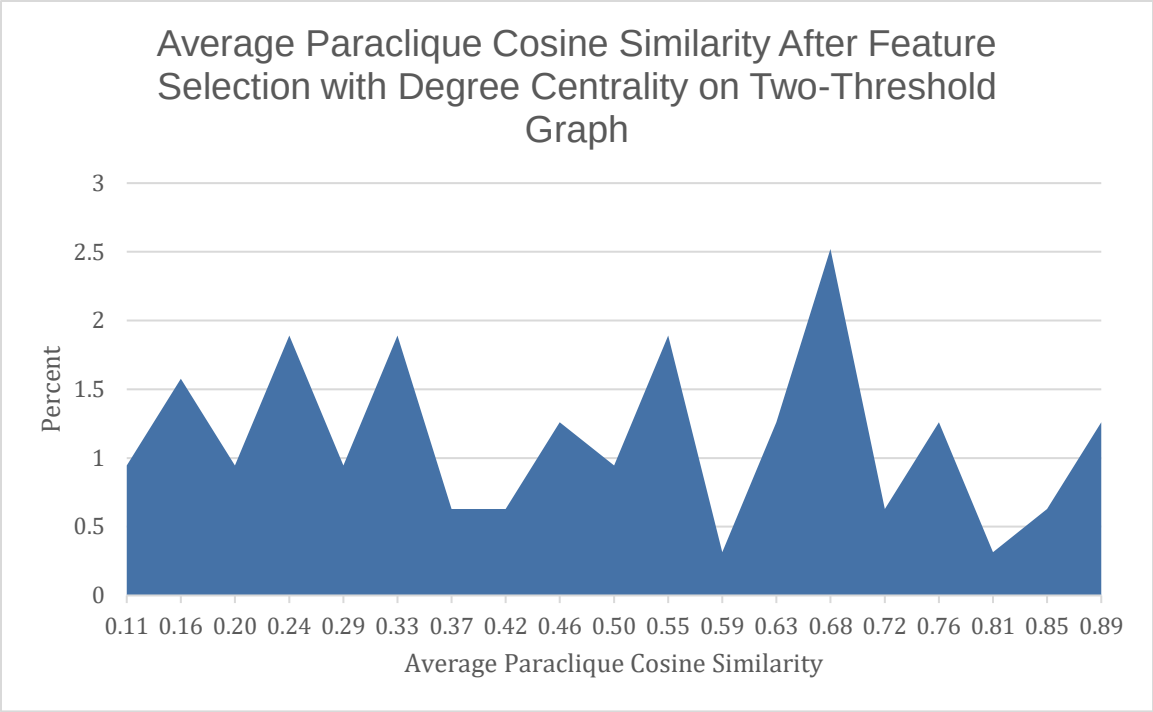


Figure 24: *The distribution of the average cosine similarity between paraclique members for all paracliques after variables were selected by Degree Centrality applied to an autocorrelates graph constructed using the two-fold threshold method. Paraclique diversity did improve with a reduction in average paraclique cosine similarity at 0.9 and an increased proportion at 0.65 and below.*

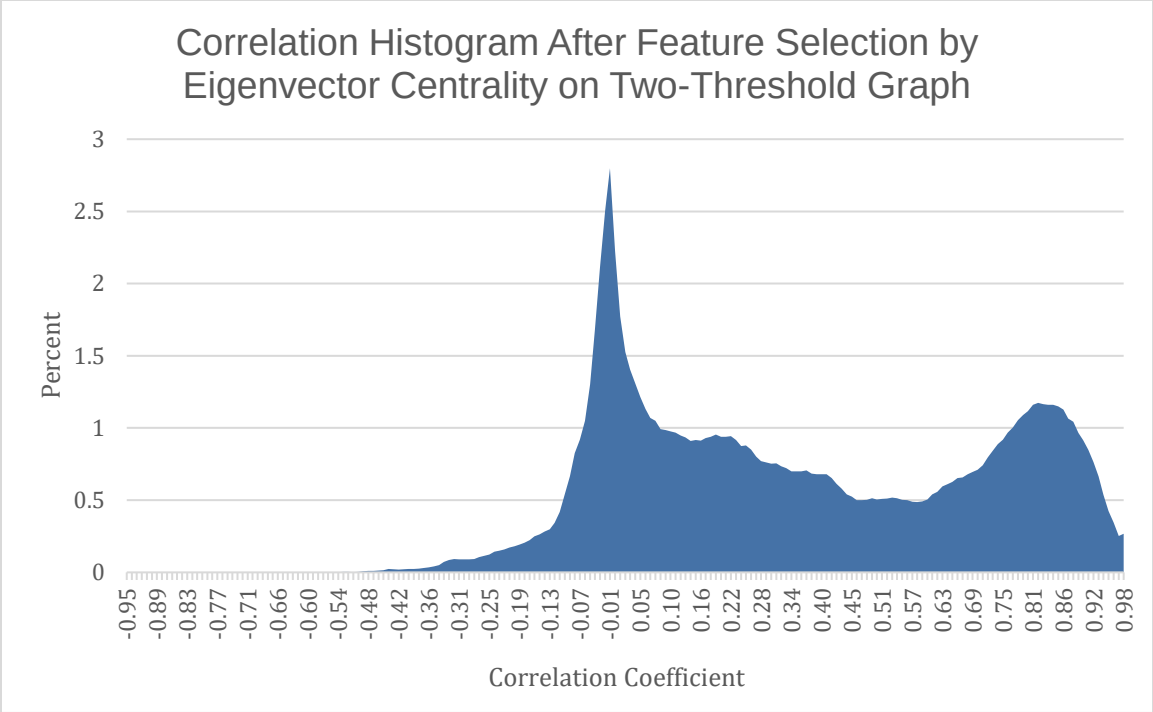


Figure 25: *The distribution of Pearson's correlation coefficients after variables were selected by eigenvector centrality applied to an autocorrelates graph constructed using the two-fold threshold method. It removed variables that shared correlation values around 0.50, increasing the proportion of values 0.90 and above to 6.1%.*

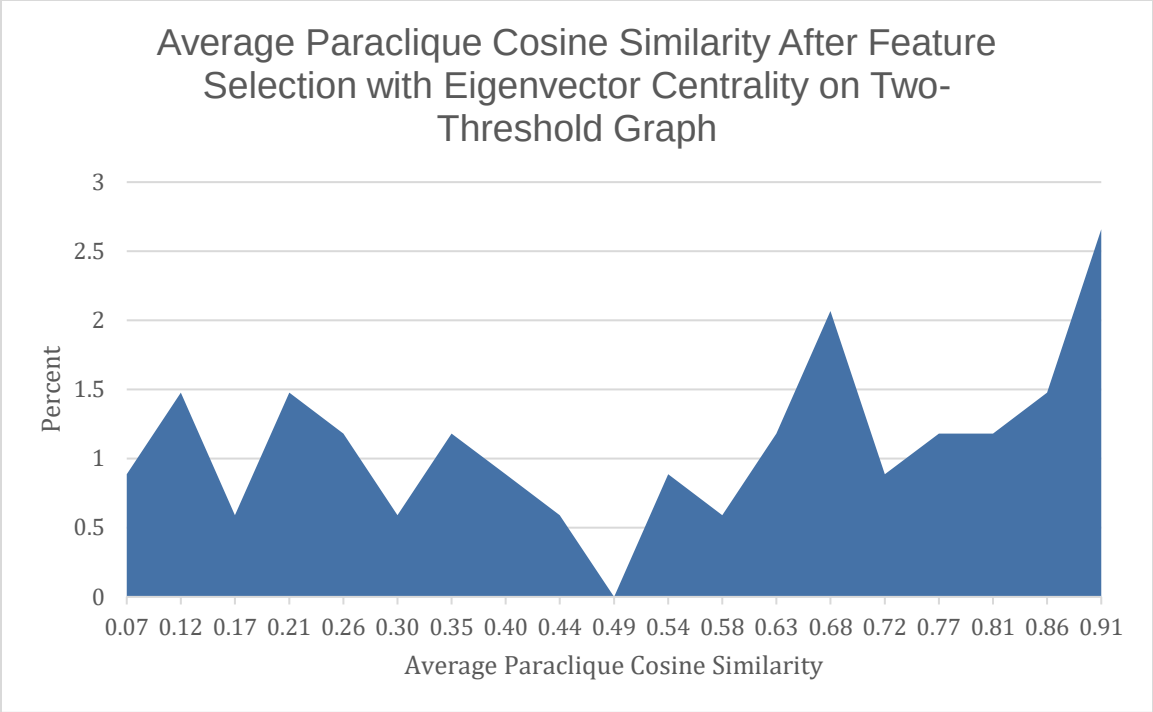


Figure 26: *The distribution of the average cosine similarity between paraclique members for all paracliques after variables were selected by eigenvector centrality applied to an autocorrelates graph constructed using the two-fold threshold method. Paraclique diversity did not improve much with such large a large proportion of correlation values greater than or equal to 0.90.*

7. CONCLUSIONS

The main contribution of this chapter is a systematic approach to the management of noise and multicollinearity as it relates to public health data, plus a novel autocorrelation reduction methodology that can ameliorate their effects. We empirically tested the effectiveness of these tools in the context of distributional skew and cluster diversity. While our focus has been on algorithmic techniques, it is important to remember that domain expertise is generally required.

Limitations of this study include sparseness of the meta data available and limits imposed on the time frames considered. Many variables in fact share the same meta data, while the omission of data accumulated before 2014 may bias results.

We foresee numerous avenues for future research. For example, logical OR could be used to replace logical AND in our two-threshold approach. Other thresholds and even other similarity metrics could be studied. And more complicated feature selection technologies, such as spectral methods, could be employed. In addition, while machine learning is currently very much in vogue, it often ignores multicollinearity at the cost of model interpretability [144], which can result in gross misassignments in feature importance [145, 146]. Finally, data curation for the PHE can be improved. Many variables shared the same description with only metadata pertaining to year and demographic signifiers differing. For example, the PHE contains variables for various types of cancer deaths, however, the PHE dictionary labels these variables simply “cancer deaths” with no concern for type of cancer. As a consequence, only the year the data was measured and demographic information could distinguish them, even if their correlations varied widely (Figure 27).

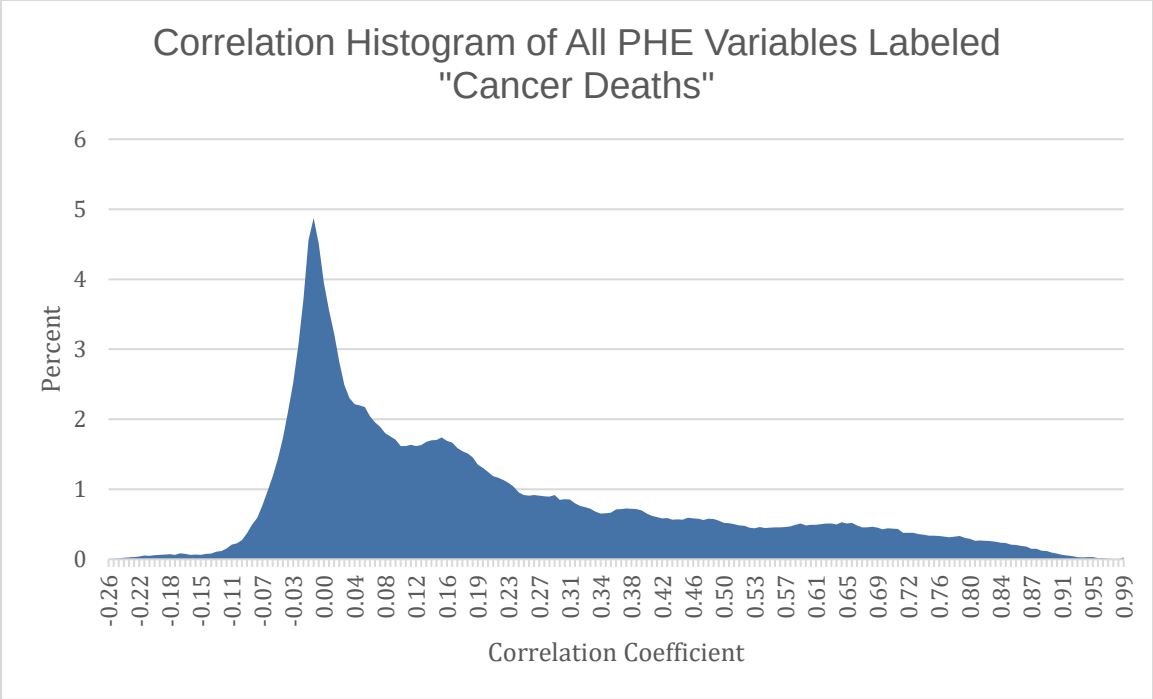


Figure 27: *The distribution of Pearson's correlation coefficients for all variables with the description "cancer deaths." Even though these variables are described the same, their correlation distribution demonstrates a wide range of variables.*

CHAPTER V
CONCLUSION

Domination has been shown to be an important concept in a diverse range of fields. In this dissertation, we have expanded the fields to which the concepts of domination are applied and developed efficient algorithms for its use in already established fields. While these projects are disjoint in their application settings, they share the commonality of data prioritization to focus further studies.

Summary of Contributions

In this dissertation we have focused our efforts on the development and implementation of MDS algorithms to reduce the complexity of high dimensional data in order to aid in the focus of their study. In Chapter 2 we developed novel algorithms that greatly improved upon those found in the literature that were used to determine vertices appropriate for controlling the entirety of a network. Contributions from this chapter are the development of five novel classification rules and two novel classification algorithms presented therein. The rules and algorithms we developed subsumed and greatly improved upon known existing techniques. We demonstrated this using a test suite of networks derived from a wide variety of biological data. In each instance, the algorithms we developed outperformed know methods in the literature.

In Chapter 3 we utilized MDS to select a subset of previously understudied genes for null allele production for the IMPC. We demonstrated the ability of our method to select a subset that is more representative of the whole than that selected by pseudorandom selection or minimal dominating set. Our method also delivered to the IMPC a systematic approach to complete a full catalog of protein coding null allele models. The method we developed in this chapter is also generalizable to

other problems in systems biology where resources are limited, and it is intractable to perform all experiments desired.

Finally in Chapter 4 we identified problems inherent to public health datasets and prescribed methods to remedy them. We developed a novel graph theoretical method based on MDS to reduce the prevalence of autocorrelates in public health datasets. Using the PHE, we tested methods described to determine their effects on correlation distributions, autocorrelation reduction, kurtosis, and paraclique membership diversity. We demonstrated the utility of our novel method by showing it reduced autocorrelation in correlation distributions and increased paraclique membership diversity.

REFERENCES

1. Cook, C.E., et al., *The European Bioinformatics Institute in 2016: Data growth and integration.* Nucleic Acids Res, 2016. **44**(D1): p. D20-6.
2. Manrai, A.K., et al., *Informatics and Data Analytics to Support Exposome-Based Discovery for Public Health.* Annu Rev Public Health, 2017. **38**: p. 279-294.
3. Garey, M.R. and D.S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. 1979, W. H. Freeman and Company. p. 1-340.
4. Downey, R.G. and M.R. Fellows, *Fixed-Parameter Tractability and Completeness I: Basic Results.* SIAM Journal on Computing, 1995. **24**: p. 873–921.
5. Kelleher, L.L. and M.B. Cozzens, *Dominating Sets in Social Network Graphs.* Mathematical Social Sciences, 1988. **16**(3): p. 267-279.
6. Nacher, J.C. and T. Akutsu, *Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control.* New Journal of Physics, 2012. **14**(7): p. 073005.
7. Nacher, J.C. and T. Akutsu, *Analysis on controlling complex networks based on dominating sets.* Ic-Msquare 2012: International Conference on Mathematical Modelling in Physical Sciences, 2013. **410**.
8. Nacher, J.C. and T. Akutsu, *Minimum dominating set-based methods for analyzing biological networks.* Methods, 2016. **102**: p. 57-63.
9. Eubank, S., et al., *Structural and algorithmic aspects of massive social networks*, in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. 2004, Society for Industrial and Applied Mathematics: New Orleans, Louisiana. p. 718–727.

10. Hagan, R.D., et al. *Towards Controllability Analysis of Dynamic Networks Using Minimum Dominating Set*. in *2020 IEEE 23rd International Conference on Information Fusion*. 2020.
11. Ravindran, V., V. Sunitha, and G. Bagler, *Identification of critical regulatory genes in cancer signaling network using controllability analysis.* Physica a-Statistical Mechanics and Its Applications, 2017. **474**: p. 134-143.
12. Schwartz, J.M., et al., *Probabilistic controllability approach to metabolic fluxes in normal and cancer tissues.* Nature Communications, 2019. **10**.
13. Wakai, R., et al., *Identification of genes and critical control proteins associated with inflammatory breast cancer using network controllability.* PLoS One, 2017. **12**(11): p. e0186353.
14. Sun, P.G., *Co-Controllability of Drug-Disease-Gene Network.* New Journal of Physics, 2015. **17**(8).
15. Bakhteh, S., A. Ghaffari-Hadigheh, and N. Chaparzadeh, *Identification of Minimum Set of Master Regulatory Genes in Gene Regulatory Networks.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2020. **17**(3): p. 999-1009.
16. Lee, B., et al., *The Hidden Control Architecture of Complex Brain Networks.* iScience, 2019. **13**: p. 154-162.
17. Wuchty, S., *Controllability in protein interaction networks.* Proceedings of the National Academy of Sciences of the United States of America, 2014. **111**(19): p. 7156-7160.
18. Zhang, X.F., et al., *Comparative analysis of housekeeping and tissue-specific driver nodes in human protein interaction networks.* BMC Bioinformatics, 2016. **17**.
19. Zhang, X.-F., et al., *Determining Minimum Set of Driver Nodes in Protein-Protein Interaction Networks.* BMC Bioinformatics, 2015. **16**: p. 146.

20. Ravindran, V., et al., *Network controllability analysis of intracellular signalling reveals viruses are actively controlling molecular systems.* Sci Rep, 2019. **9**(1): p. 2066.
21. Kagami, H., et al., *Determining Associations between Human Diseases and Non-Coding RNAs with Critical Roles in Network Control.* Scientific Reports, 2015. **5**: p. 14577.
22. Cacheiro, P., et al., *New models for human disease from the International Mouse Phenotyping Consortium.* Mammalian genome : official journal of the International Mammalian Genome Society, 2019. **30**(5-6): p. 143-150.
23. Perkins, A.D. and M.A. Langston, *Threshold selection in gene co-expression networks using spectral graph theory techniques.* BMC Bioinformatics, 2009. **10**(11): p. S4.
24. Grady, S.K., et al., *Domination based classification algorithms for the controllability analysis of biological interaction networks.* Scientific Reports, 2022. **12**(1): p. 11897.
25. Fomin, F.V., et al. *Bounding the Number of Minimal Dominating Sets: A Measure and Conquer Approach*. 2005. Berlin, Heidelberg: Springer Berlin Heidelberg.
26. Nacher, J.C. and T. Akutsu, *Analysis of critical and redundant nodes in controlling directed and undirected complex networks using dominating sets.* Journal of Complex Networks, 2014. **2**(4): p. 394-412.
27. Ishitsuka, M., T. Akutsu, and J.C. Nacher, *Critical controllability in proteome-wide protein interaction network integrating transcriptome.* Scientific Reports, 2016. **6**.
28. Ostrowski, J., et al., *Orbital branching.* Mathematical Programming, 2011. **126**(1): p. 147-178.
29. Miyazaki, T., *The Complexity of McKay's Canonical Labeling Algorithm.* DIMACS Series in Discrete Mathematicsand Theoretical Computer Science, 1997. **28**: p. 239-256.

30. Junttila, T. and P. Kaski. *Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs*. in *Proceedings, Workshop on Algorithm Engineering and Experiments*. 2007. New Orleans, Louisiana: SIAM.
31. McKay, B.D. and A. Piperno, *Practical graph isomorphism, II.* Journal of Symbolic Computation, 2014. **60**: p. 94-112.
32. H. Katebi, K.A. Sakallah, and I.L. Markov. *Symmetry and Satisfiability: An Update*. in *Proceedings, International Conference on Theory and Applications of Satisfiability Testing*. 2010. Edinburgh, Scotland: Springer LNCS.
33. *CPLEX Optimization Studio*. 2021; Available from: https://www.ibm.com.
34. *Gurobi Optimizer*. 2021; Available from: https://www.gurobi.com.
35. Iwata, Y. *A Faster Algorithm for Dominating Set Analyzed by the Potential Method*. in *International Conference on Parameterized and Exact Computation*. 2011. Saarbrücken, Germany: Springer.
36. Berge, C., *Theory of Graphs and its Applications*. 1962, London: Methuen Publishing.
37. Sampathkumar, E. and H.B. Walikar, *The connected domination number of a graph.* Math. Phys. Sci., 1979. **13**(6): p. 607–613.
38. Cockayne, E.J., R.M. Dawes, and S.T. Hedetniemi, *Total domination in graphs.* Networks, 1980. **10**: p. 211–219.
39. Broido, A.D. and A. Clauset, *Scale-Free Networks are Rare.* Nature Communications, 2019. **10**.
40. Marinka Zitnik, R.S., Sagar Maheshwari, and Jure Leskovec. *BioSNAP Datasets: Stanford Biomedical Network Dataset Collection*. 2018.
41. Pratt, D., et al., *NDEx, the Network Data Exchange.* Cell Systems, 2015. **1**(4): p. 302-305.
42. Barrett, T., et al., *NCBI GEO: archive for functional genomics data sets--update.* Nucleic acids research, 2013. **41**(Database issue): p. D991-D995.
43. Pei, H., et al., *FKBP51 affects cancer cell response to chemotherapy by negatively regulating Akt.* Cancer cell, 2009. **16**(3): p. 259-266.

44. Ellsworth, K.A., et al., *Contribution of FKBP5 genetic variation to gemcitabine treatment and survival in pancreatic adenocarcinoma.* PloS one, 2013. **8**(8): p. e70216-e70216.
45. Li, L., et al., *Genetic variations associated with gemcitabine treatment outcome in pancreatic cancer.* Pharmacogenet Genomics, 2016. **26**(12): p. 527-537.
46. Hong, Y., et al., *A 'metastasis-prone' signature for early-stage mismatch-repair proficient sporadic colorectal cancer patients and its implications for possible therapeutics.* Clin Exp Metastasis, 2010. **27**(2): p. 83-90.
47. Pedraza, V., et al., *Gene expression signatures in breast cancer distinguish phenotype characteristics, histologic subtypes, and tumor invasiveness.* Cancer, 2010. **116**(2): p. 486-96.
48. Zheng, B., et al., *PGC-1α, a potential therapeutic target for early intervention in Parkinson's disease.* Sci Transl Med, 2010. **2**(52): p. 52ra73.
49. Baker, E., et al., *GeneWeaver: data driven alignment of cross-species genomics in biology and disease.* Nucleic Acids Research, 2016. **44**(D1): p. D555-D559.
50. Pitkänen, J.P., et al., *Excess mannose limits the growth of phosphomannose isomerase PMI40 deletion strain of Saccharomyces cerevisiae.* J Biol Chem, 2004. **279**(53): p. 55737-43.
51. Rossi, R.A. and N.K. Ahmed. *The Network Data Repository with Interactive Graph Analytics and Visualization*. in *Proceedings, AAAI Conference on Artificial Intelligence*. 2015. Austin, Texas.
52. Yu, H., et al., *High-quality binary protein interaction map of the yeast interactome network.* Science (New York, N.Y.), 2008. **322**(5898): p. 104-110.
53. Oughtred, R., et al., *The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions.* Protein science : a publication of the Protein Society, 2021. **30**(1): p. 187-200.

54. Luck, K., et al., *A reference map of the human binary protein interactome.* Nature, 2020. **580**(7803): p. 402-408.
55. Zhang, B. and S. Horvath, *A general framework for weighted gene co-expression network analysis.* Stat Appl Genet Mol Biol, 2005. **4**: p. Article17.
56. Ziebarth, J.D. and Y. Cui, *Precise Network Modeling of Systems Genetics Data Using the Bayesian Network Webserver*, in *Systems Genetics: Methods and Protocols*, K. Schughart and R.W. Williams, Editors. 2017, Springer New York: New York, NY. p. 319-335.
57. Song, M.J., et al., *Reconstructing generalized logical networks of transcriptional regulation in mouse brain from temporal gene expression data.* EURASIP journal on bioinformatics & systems biology, 2009. **2009**(1): p. 545176-545176.
58. Vafaee, F., et al., *Novel semantic similarity measure improves an integrative approach to predicting gene functional associations.* BMC Syst Biol, 2013. **7**: p. 22.
59. Peng, J., W. Hui, and X. Shang, *Measuring phenotype-phenotype similarity through the interactome.* BMC Bioinformatics, 2018. **19**(5): p. 114.
60. Wei, D.H., et al., *Construction of Disease Similarity Networks Using Concept Embedding and Ontology.* Studies in health technology and informatics, 2019. **264**: p. 442-446.
61. Halu, A., et al., *The multiplex network of human diseases.* NPJ Syst Biol Appl, 2019. **5**: p. 15.
62. Wolfe, C.J., I.S. Kohane, and A.J. Butte, *Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks.* BMC bioinformatics, 2005. **6**: p. 227-227.
63. Pandey, A.K., et al., *Functionally enigmatic genes: a case study of the brain ignorome.* PloS one, 2014. **9**(2): p. e88889-e88889.

64. Riba, M., et al., *Revealing the acute asthma ignorome: characterization and validation of uninvestigated gene networks.* Sci Rep, 2016. **6**: p. 24647.
65. Stoeger, T., et al., *Large-scale investigation of the reasons why potentially important genes are ignored.* PLoS Biol, 2018. **16**(9): p. e2006643.
66. Stoeger, T. and L.A. Nunes Amaral, *COVID-19 research risks ignoring important host genes due to pre-established research patterns.* eLife, 2020. **9**: p. e61981.
67. Karczewski, K.J., et al., *The mutational constraint spectrum quantified from variation in 141,456 humans.* Nature, 2020. **581**(7809): p. 434-443.
68. Reynolds, T., et al., *Finding human gene-disease associations using a Network Enhanced Similarity Search (NESS) of multi-species heterogeneous functional genomics data.* 2020: p. 2020.03.11.987552.
69. *The Gene Ontology resource: enriching a GOld mine.* Nucleic Acids Res, 2021. **49**(D1): p. D325-d334.
70. Ashburner, M., et al., *Gene Ontology: tool for the unification of biology.* Nature Genetics, 2000. **25**(1): p. 25-29.
71. Baker, E.J., et al., *GeneWeaver: a web-based system for integrative functional genomics.* Nucleic acids research, 2012. **40**(Database issue): p. D1067-D1076.
72. Szklarczyk, D., et al., *STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets.* Nucleic Acids Research, 2018. **47**(D1): p. D607-D613.
73. Motenko, H., et al., *MouseMine: a new data warehouse for MGI.* Mamm Genome, 2015. **26**(7-8): p. 325-30.
74. Jaccard, P., *THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1.* 1912. **11**(2): p. 37-50.
75. Van Rossum, G.a.D., Fred L., *Python 3 Reference Manual*. 2009: CreateSpace.

76. Matsumoto, M. and T. Nishimura, *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator.* 1998. **8**(1 %J ACM Trans. Model. Comput. Simul.): p. 3–30.
77. Peterson, K.A. and S.A. Murray, *Progress towards completing the mutant mouse null resource.* Mamm Genome, 2021.
78. Birling, M.-C., et al., *A resource of targeted mutant mouse lines for 5,061 genes.* Nature genetics, 2021. **53**(4): p. 416-419.
79. Garey, M.R. and D.S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. 1990: W. H. Freeman & Co.
80. Chvatal, V., *A Greedy Heuristic for the Set-Covering Problem.* Mathematics of Operations Research, 1979. **4**(3): p. 233-235.
81. Wild, C.P., *Complementing the genome with an "exposome": the outstanding challenge of environmental exposure measurement in molecular epidemiology.* Cancer Epidemiol Biomarkers Prev, 2005. **14**(8): p. 1847-50.
82. Juarez, P.D., et al., *The public health exposome: a population-based, exposure science approach to health disparities research.* International journal of environmental research and public health, 2014. **11**(12): p. 12866-12895.
83. Signorello, L.B., M.K. Hargreaves, and W.J. Blot, *The Southern Community Cohort Study: investigating health disparities.* Journal of health care for the poor and underserved, 2010. **21**(1 Suppl): p. 26-37.
84. Wishart, D., et al., *T3DB: the toxic exposome database.* Nucleic acids research, 2015. **43**(Database issue): p. D928-D934.
85. Davis, A.P., et al., *Comparative Toxicogenomics Database (CTD): update 2021.* Nucleic Acids Research, 2020. **49**(D1): p. D1138-D1143.
86. Maitre, L., et al., *Human Early Life Exposome (HELIX) study: a European population-based exposome cohort.* 2018. **8**(9): p. e021311.

87. Neveu, V., et al., *Exposome-Explorer: a manually-curated database on biomarkers of exposure to dietary and environmental factors.* Nucleic acids research, 2017. **45**(D1): p. D979-D984.
88. Lakerveld, J., et al., *Deep phenotyping meets big data: the Geoscience and hEalth Cohort COnsortium (GECCO) data to enable exposome studies in The Netherlands.* International Journal of Health Geographics, 2020. **19**(1): p. 49.
89. Stingone, J.A., et al., *Toward Greater Implementation of the Exposome Research Paradigm within Environmental Epidemiology.* Annual review of public health, 2017. **38**: p. 315-327.
90. Sun, Z., et al., *Statistical strategies for constructing health risk models with multiple pollutants and their interactions: possible choices and comparisons.* Environmental Health, 2013. **12**(1): p. 85.
91. *The problem of multicollinearity*, in *Understanding Regression Analysis*. 1997, Springer US: Boston, MA. p. 176-180.
92. Graham, M.H., *CONFRONTING MULTICOLLINEARITY IN ECOLOGICAL MULTIPLE REGRESSION.* 2003. **84**(11): p. 2809-2815.
93. *Environmental Protection Agency Open Data Portal*. 2020; Available from: https://www.epa.gov/data.
94. The Center for the Evaluative Clinical Sciences, D.M.S., *The Dartmouth atlas of health care*. 1996: Chicago, Ill. : American Hospital Publishing, [1996] ©1996.
95. *U.S. Census Bureau. Data Portal.* 2020.
96. Valdez, R.B., et al., *Association of Cardiovascular Disease and Long-Term Exposure to Fine Particulate Matter (PM2.5) in the Southeastern United States.* 2021. **12**(8): p. 947.
97. Gittner, L.S., et al., *A multifactorial obesity model developed from nationwide public health exposome data and modern computational analyses.* Obes Res Clin Pract, 2017. **11**(5): p. 522-533.

98. Juarez, P., et al., *A novel approach to analyzing lung cancer mortality disparities: Using the exposome and a graph-theoretical toolchain.* 2017. **2**(2): p. 33-44.
99. Kershenbaum, A.D., et al., *Exploration of preterm birth rates using the public health exposome database and computational analysis methods.* International journal of environmental research and public health, 2014. **11**(12): p. 12346-12366.
100. Juarez, P.D., et al., *Use of an Exposome Approach to Understand the Effects of Exposures From the Natural, Built, and Social Environments on Cardio-Vascular Disease Onset, Progression, and Outcomes.* Front Public Health, 2020. **8**: p. 379.
101. Van den Broeck, J., et al., *Data cleaning: detecting, diagnosing, and editing data abnormalities.* PLoS Med, 2005. **2**(10): p. e267.
102. Jakobsen, J.C., et al., *When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts.* BMC Medical Research Methodology, 2017. **17**(1): p. 162.
103. Tiwari, C., K. Beyer, and G. Rushton, *The Impact of Data Suppression on Local Mortality Rates: The Case of CDC WONDER.* 2014. **104**(8): p. 1386-1388.
104. Nogueira, S. and G. Brown. *Measuring the Stability of Feature Selection*. 2016. Cham: Springer International Publishing.
105. Beraha, M., et al. *Feature Selection via Mutual Information: New Theoretical Insights*. in *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019.
106. Frénay, B., G. Doquire, and M. Verleysen, *Is mutual information adequate for feature selection in regression?* Neural Networks, 2013. **48**: p. 1-7.
107. Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32.
108. Basu, S., et al., *Iterative random forests to discover predictive and stable high-order interactions.* 2018. **115**(8): p. 1943-1948.

109. Chen, T. and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, Association for Computing Machinery: San Francisco, California, USA. p. 785–794.
110. Menze, B.H., et al., *A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data.* BMC Bioinformatics, 2009. **10**(1): p. 213.
111. Nicodemus, K.K., et al., *The behaviour of random forest permutation-based variable importance measures under predictor correlation.* BMC Bioinformatics, 2010. **11**(1): p. 110.
112. Lundberg, S.M. and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, Curran Associates Inc.: Long Beach, California, USA. p. 4768–4777.
113. Ohanyan, H., et al., *Machine learning approaches to characterize the obesogenic urban exposome.* Environment International, 2022. **158**: p. 107015.
114. Guyon, I., et al., *Gene Selection for Cancer Classification using Support Vector Machines.* Machine Learning, 2002. **46**(1): p. 389-422.
115. Darst, B.F., K.C. Malecki, and C.D. Engelman, *Using recursive feature elimination in random forest to account for correlated variables in high dimensional data.* BMC genetics, 2018. **19**(Suppl 1): p. 65-65.
116. Cai, J., et al., *Feature selection in machine learning: A new perspective.* Neurocomputing, 2018. **300**: p. 70-79.
117. Dy, J.G. and C.E. Brodley, *Feature Selection for Unsupervised Learning.* 2004. **5**: p. 845–889.
118. Langston, M.A., et al., *Scalable combinatorial tools for health disparities research.* International journal of environmental research and public health, 2014. **11**(10): p. 10419-10443.

119. Solorio-Fernández, S., J.A. Carrasco-Ochoa, and J.F. Martínez-Trinidad, *A review of unsupervised feature selection methods.* Artificial Intelligence Review, 2020. **53**(2): p. 907-948.
120. Zhang, Z. and E.R. Hancock. *A Graph-Based Approach to Feature Selection*. 2011. Berlin, Heidelberg: Springer Berlin Heidelberg.
121. Schroeder, D.T., et al. *Graph-based Feature Selection Filter Utilizing Maximal Cliques*. in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 2019.
122. Henni, K., N. Mezghani, and C. Gouin-Vallerand, *Unsupervised graph-based feature selection via subspace and pagerank centrality.* Expert Systems with Applications, 2018. **114**: p. 46-53.
123. Moradi, P. and M. Rostami, *A graph theoretic approach for unsupervised feature selection.* Engineering Applications of Artificial Intelligence, 2015. **44**: p. 33-45.
124. Giarelis, N., N. Kanakaris, and N. Karacapilidis, *An Innovative Graph-Based Approach to Advance Feature Selection from Multiple Textual Documents.* Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I, 2020. **583**: p. 96-106.
125. Das, A.K., et al., *An information-theoretic graph-based approach for feature selection.* Sādhanā, 2019. **45**(1): p. 11.
126. He, X., D. Cai, and P. Niyogi, *Laplacian score for feature selection*, in *Proceedings of the 18th International Conference on Neural Information Processing Systems*. 2005, MIT Press: Vancouver, British Columbia, Canada. p. 507–514.
127. Santos, S., et al., *Applying the exposome concept in birth cohort research: a review of statistical approaches.* European journal of epidemiology, 2020. **35**(3): p. 193-204.

128. Sparck Jones, K., *A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL.* Journal of Documentation, 1972. **28**(1): p. 11-21.
129. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python.* 2011. **12**: p. 2825-2830.
130. Singhal, A., *Modern Information Retrieval: A Brief Overview.* IEEE Data Eng. Bull., 2001. **24**(4): p. 35-43.
131. John Forrest, T.R., Haroldo Gambini Santos, Stefan Vigerske, Lou Hafer, John Forrest, Bjarni Kristjansson, jpfasano, EdwinStraver, Miles Lubin, rlougee, jpgoncal1, Jan-Willem, h-i-gassmann, Samuel Brito, Cristina, Matthew Saltzman, tosttost, Fumiaki MATSUSHIMA, *coin-or/Cbc: Release releases/2.10.7 (releases/2.10.7).* 2022, Zenodo.
132. Túlio A. M. Toffolo, H.G.S., *Python MIP*. 2021: https://docs.python-mip.com/en/latest/index.html.
133. Hagberg, A., P. Swart, and D. S Chult, *Exploring network structure, dynamics, and function using NetworkX*. 2008, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
134. NERSC. *National Energy Research Scientific Computing Center*. 2020; Available from: http://www.nersc.gov/.
135. Foss, S.S.S.G., D. Korshunov, and S. Zachary, *An introduction to heavy-tailed and subexponential distributions. 1st ed*. Vol. 38. 2011.
136. Virtanen, P., et al., *SciPy 1.0: fundamental algorithms for scientific computing in Python.* Nature Methods, 2020. **17**(3): p. 261-272.
137. Chesler, E.J. and M.A. Langston. *Combinatorial Genetic Regulatory Network Analysis Tools for High Throughput Transcriptomic Data*. in *Systems Biology and Regulatory Genomics*. 2006. Berlin, Heidelberg: Springer Berlin Heidelberg.
138. Freeman, L.C., *A Set of Measures of Centrality Based on Betweenness.* Sociometry, 1977. **40**(1): p. 35-41.

139. Valenzuela, J.F.B., et al., *Degree and centrality-based approaches in network-based variable selection: Insights from the Singapore Longitudinal Aging Study.* PLOS ONE, 2019. **14**(7): p. e0219186.
140. Brin, S. and L. Page, *The anatomy of a large-scale hypertextual Web search engine.* Computer Networks and ISDN Systems, 1998. **30**(1): p. 107-117.
141. Sharma, D. and A. Surolia, *Degree Centrality*, in *Encyclopedia of Systems Biology*, W. Dubitzky, et al., Editors. 2013, Springer New York: New York, NY. p. 558-558.
142. Roffo, G. and S. Melzi. *Feature Selection via Eigenvector Centrality*. 2016.
143. Bonacich, P.J.A.J.o.S., *Power and Centrality: A Family of Measures.* 1987. **92**: p. 1170 - 1182.
144. De Veaux, R.D. and L.H. Ungar. *Multicollinearity: A tale of two nonparametric regressions*. 1994. New York, NY: Springer New York.
145. Tolosi, L. and T. Lengauer, *Classification with correlated features: unreliability of feature ranking and solutions.* Bioinformatics, 2011. **27**(14): p. 1986-94.
146. Strobl, C., et al., *Conditional variable importance for random forests.* BMC Bioinformatics, 2008. **9**(1): p. 307.

VITA

Stephen Grady was born in Fort Rucker, Alabama and shortly after moved with this mother to Gravette, Arkansas. After Graduating from Gravette High, he attended the University of Arkansas where he received a Bachelor of Science in Chemistry. Afterward, at the University of Arkansas, he worked in the labs of Dr. Woodrow Shew and then Dr. Jin-Woo Kim under the tutelage of George Sakhel. Shortly after, he was accepted to the Genome Science and Technology Program at the University of Tennessee where he studies under Dr. Michael Langston.