




2019

## COMPUTATIONAL TOOLS FOR THE DYNAMIC CATEGORIZATION AND AUGMENTED UTILIZATION OF THE GENE ONTOLOGY

Eugene Waverly Hinderer III

University of Kentucky, ehinderer01@gmail.com

Author ORCID Identifier:

 <https://orcid.org/0000-0001-8045-7083>

Digital Object Identifier: <https://doi.org/10.13023/etd.2019.362>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

### Recommended Citation

Hinderer, Eugene Waverly III, "COMPUTATIONAL TOOLS FOR THE DYNAMIC CATEGORIZATION AND AUGMENTED UTILIZATION OF THE GENE ONTOLOGY" (2019). *Theses and Dissertations--Molecular and Cellular Biochemistry*. 43.

[https://uknowledge.uky.edu/biochem\\_etds/43](https://uknowledge.uky.edu/biochem_etds/43)

This Doctoral Dissertation is brought to you for free and open access by the Molecular and Cellular Biochemistry at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Molecular and Cellular Biochemistry by an authorized administrator of UKnowledge. For more information, please contact [UKnowledge@lsv.uky.edu](mailto:UKnowledge@lsv.uky.edu).

## **STUDENT AGREEMENT:**

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

## **REVIEW, APPROVAL AND ACCEPTANCE**

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Eugene Waverly Hinderer III, Student

Dr. Hunter N. B. Moseley, Major Professor

Dr. Trevor P. Creamer, Director of Graduate Studies

COMPUTATIONAL TOOLS FOR THE DYNAMIC CATEGORIZATION AND  
AUGMENTED UTILIZATION OF THE GENE ONTOLOGY

---

DISSERTATION

---

A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy in the  
College of Medicine  
at the University of Kentucky

By  
Eugene Waverly Hinderer III  
Lexington, Kentucky  
Director: Dr. Hunter N. B. Moseley, Professor of Molecular & Cellular Biochemistry  
Lexington, Kentucky  
2019

Copyright © Eugene Waverly Hinderer III 2019  
<https://orcid.org/0000-0001-8045-7083>

## ABSTRACT OF DISSERTATION

### COMPUTATIONAL TOOLS FOR THE DYNAMIC CATEGORIZATION AND AUGMENTED UTILIZATION OF THE GENE ONTOLOGY

Ontologies provide an organization of language, in the form of a network or graph, which is amenable to computational analysis while remaining human-readable. Although they are used in a variety of disciplines, ontologies in the biomedical field, such as Gene Ontology, are of interest for their role in organizing terminology used to describe—among other concepts—the functions, locations, and processes of genes and gene-products. Due to the consistency and level of automation that ontologies provide for such annotations, methods for finding enriched biological terminology from a set of differentially identified genes in a tissue or cell sample have been developed to aid in the elucidation of disease pathology and unknown biochemical pathways. However, despite their immense utility, biomedical ontologies have significant limitations and caveats. One major issue is that gene annotation enrichment analyses often result in many redundant, individually enriched ontological terms that are highly specific and weakly justified by statistical significance. These large sets of weakly enriched terms are difficult to interpret without manually sorting into appropriate functional or descriptive categories. Also, relationships that organize the terminology within these ontologies do not contain descriptions of semantic scoping or scaling among terms. Therefore, there exists some ambiguity, which complicates the automation of categorizing terms to improve interpretability.

We emphasize that existing methods enable the danger of producing incorrect mappings to categories as a result of these ambiguities, unless simplified and incomplete versions of these ontologies are used which omit problematic relations. Such ambiguities could have a significant impact on term categorization, as we have calculated upper boundary estimates of potential false categorizations as high as 121,579 for the misinterpretation of a single scoping relation, *has\_part*, which accounts for approximately 18% of the total possible mappings between terms in the Gene Ontology. However, the omission of problematic relationships results in a significant loss of retrievable information. In the Gene Ontology, this accounts for a 6% reduction for the omission of a single relation. However, this percentage should increase drastically when considering all relations in an ontology. To address these issues, we have developed methods which categorize individual ontology terms into broad, biologically-related concepts to improve the interpretability and statistical significance of gene-annotation enrichment studies, meanwhile addressing the lack of semantic scoping and scaling descriptions among ontological relationships so that annotation enrichment analyses can be performed across a more complete representation of the ontological graph.

We show that, when compared to similar term categorization methods, our method produces categorizations that match hand-curated ones with similar or better accuracy, while not requiring the user to compile lists of individual ontology term IDs. Furthermore,

our handling of problematic relations produces a more complete representation of ontological information from a scoping perspective, and we demonstrate instances where medically-relevant terms--and by extension putative gene targets--are identified in our annotation enrichment results that would be otherwise missed when using traditional methods. Additionally, we observed a marginal, yet consistent improvement of statistical power in enrichment results when our methods were used, compared to traditional enrichment analyses that utilize ontological ancestors. Finally, using scalable and reproducible data workflow pipelines, we have applied our methods to several genomic, transcriptomic, and proteomic collaborative projects.

KEYWORDS: Annotation Enrichment Analysis, Biomedical Ontologies, Information Retrieval, Ontological Maintenance, Semantic Correspondence

Eugene Waverly Hinderer III

---

*(Name of Student)*

08/05/2019

---

Date

COMPUTATIONAL TOOLS FOR THE DYNAMIC CATEGORIZATION AND  
AUGMENTED UTILIZATION OF THE GENE ONTOLOGY

By  
Eugene Waverly Hinderer III

Dr. Hunter N. B. Moseley  
\_\_\_\_\_  
Director of Dissertation

Trevor P. Creamer  
\_\_\_\_\_  
Director of Graduate Studies

08/05/2019  
\_\_\_\_\_

Date

## DEDICATION

To Minerva, who helped in the way that a cat can.

## ACKNOWLEDGMENTS

The following dissertation, while an individual work, benefited from the insights and direction of several people. First, my Dissertation Chair, Dr. Hunter N.B. Moseley, exemplifies the high-quality scholarship to which I aspire. In addition, Dr. Robert Flight provided timely and instructive comments, evaluation, and advice at every stage of the dissertation process, allowing me to complete this project on schedule. Furthermore, I would also like to thank the members of my laboratory—past and present—Dr. Joshua Mitchell, Dr. Sen Yao, Dr. Andrey Smelter, Dr. Xi ‘Bill’ Chen, Dr. Thilakam Murali, Shruti Sinha, Kelly Sovacool, Patrick ‘Kai’ Baker, Huan Jin, and Christian Powel, all of whom provided support throughout my graduate school career. Next, I wish to thank the complete Dissertation Committee, and outside reader, respectively: Dr. Yvonne Fondufe-Mittendorf, Dr. Kathleen O’Connor, Dr. Chi Wang, and Dr. Jin Chen. Each individual provided insights that guided and challenged my thinking, substantially improving the finished product.

In addition to the technical and scientific assistance above, I received equally important assistance from family. My Fiancée, Dr. Marisa Kamelgarn, provided on-going support throughout the dissertation process, as well as useful advice, critical for scheduling and preparing for my PhD defense. Finally, I wish to thank my parents and the rest of my family, who graciously supported me as a first-generation college graduate, without their support I would not have had the opportunity to be where I am today.

# TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	iii
LIST OF TABLES .....	viii
LIST OF FIGURES .....	ix
LIST OF ADDITIONAL FILES .....	x
CHAPTER 1. Introduction.....	1
1.1 ..... <i>Ontologies and Their Role in Modern Scientific Research</i>	1
1.2 ..... <i>The Gene Ontology (GO)</i>	4
1.2.1 Overview .....	4
1.2.2 Data structure.....	5
1.2.3 Gene Ontology Annotations .....	8
1.3 ..... <i>Annotation Enrichment and the Importance of Ontological Inference</i>	9
1.4 ..... <i>Difficulty in Representing Biological Concepts Derived from Omics-Level Research</i>	11
1.5 ..... <i>Term Categorization Approaches</i>	12
1.6 ..... <i>Semantic Similarity in the Context of Broad Term Categorization</i>	13
1.7 ..... <i>Maintenance of Ontologies</i>	14
1.8 ..... <i>Path Traversal Issues in GO</i>	15
1.9 ..... <i>Axiomatic Versus Semantic Scoping Interpretation of Mereological Relations in GO</i>	16
CHAPTER 2. Materials and Methods .....	19
2.1 ..... <i>The Gene Ontology Categorization Suite (GOcats)</i>	19
2.1.1 Methodological Overview and Design Rationale for GOcats.....	19
2.1.2 GOcats Implementation Overview.....	22
2.1.3 GOcats Specific Implementation Details .....	24
2.1.4 Defining and Traversing Categorization-relevant Edges in GO .....	27

2.2 .....	<i>Pipelines Incorporating GOcats' Ancestor Paths and Categorizations into Annotation Enrichment Analyses.....</i>	28
2.3 .....	<i>Visualizing Protein-Protein Interaction Network Visualizations based on Enrichment Results .....</i>	32
CHAPTER 3.	GOcats: A tool for categorizing Gene Ontology into subgraphs of user-defined concepts..	37
3.1 .....	<i>Background .....</i>	37
3.2 .....	<i>Results .....</i>	39
3.2.1	GOcats Compactly Organizes GO Subcellular Localization Terms into User-Specified Categories .....	39
3.2.2	GOcats-derived Category Subgraphs Compare Well with Similar Subgraphs Derived by Other Methods.....	41
3.2.3	Custom-tailoring of GO Slim-like Categories with GOcats Allows for Robust Knowledgebase Gene Annotation Mining.....	43
3.3 .....	<i>Discussion and Conclusions .....</i>	46
3.4 .....	<i>Methods .....</i>	51
3.4.1	Creating Category Mappings from UniProt's Subcellular Location Controlled Vocabulary..	51
3.4.2	Creating Category Mappings from Map2Slim.....	52
3.4.3	Mapping Gene Annotations to User-defined Categories .....	52
3.4.4	Visualizing and Characterizing Intersections of Category Subgraphs.....	53
3.4.5	Assigning Generalized Subcellular Locations to Genes from the Knowledgebase and Comparing Assignments to Experimentally-Determined Locations .....	53
3.4.6	Running Time Tests between GOcats and Map2Slim Categorizations.....	55
CHAPTER 4.	Advances in Gene Ontology Utilization Improve Statistical Power of Annotation Enrichment	75
4.1 .....	<i>Background .....</i>	75
4.2 .....	<i>Results .....</i>	78

4.2.1	GOcats' Reinterpretation of the <i>has_part</i> Relation Increases the Information Retrieval from GO and Avoids Potential Misinterpretations of Ambiguous Relationship Inferences .....	78
4.2.2	GOcats' Reinterpretation of the <i>has_part</i> Relations Provides Improved Annotation Enrichment Statistical Power .....	80
4.3	..... <i>Discussion</i>	
	.....	82
4.3.1	Issues with Semantic Correspondence.....	82
4.3.2	Using GOcats for Annotation Enrichment .....	86
4.4	..... <i>Conclusions</i>	
	.....	88
4.5	..... <i>Methods</i>	
	.....	89
4.5.1	Evaluating Hypothetical False Mapping and True Mapping Pairs in GO Involving the <i>has_part</i> Relation .....	89
4.5.2	Evaluating Hypothetical False Mappings Encountered When the Unaltered <i>has_part</i> Relation is Parsed with Map2Slim .....	90
4.5.3	Comparing Mapping Functionality between the Java and Perl Versions of Map2Slim .....	91
4.5.4	Annotation Enrichment Analysis of Breast Cancer Dataset.....	92
4.5.5	Annotation Enrichment of Equine Cartilage Development Dataset.....	93
4.5.6	RNASeq Analysis of Equine Cartilage Development Time Points.....	96
CHAPTER 5.	Annotation Enrichment Analysis Applications.....	110
5.1	..Identifying Enriched Annotations and Putative Gene Targets among Differentially-expressed Genes during the Fetal Developmental Progression of Equine Tissue .....	110
5.1.1	Background and Experimental Design.....	110
5.1.2	Results.....	112
5.2	..... <i>Determining Features Unique to Kentucky Lung Adenocarcinoma Mutational Profiles.</i>	
	.....	114
5.2.1	Background and Experimental Design.....	114
5.2.2	Results.....	116
CHAPTER 6.	Future Directions .....	130

6.1 .....	<i>Developing Heuristics to Automatically Assign Semantic Scaling and Scoping Correspondences between Annotation Terms Connected by Relationships in GO and Other Ontologies .....</i>	131
6.1.1	Defining relationship correspondence classes.....	132
6.1.2	Parsing and classifying relationships in the Relations Ontology. ....	132
6.1.3	Justification.....	134
6.1.4	Expected outcomes.....	135
6.2 .....	<i>Developing Algorithms That Automatically Identify Compactly-represented Concepts in GO and Other Ontologies.....</i>	136
6.2.1	Defining ontological concept compactness. ....	136
6.2.2	Automatic enumeration of ontological concepts via lexical analysis. ....	137
6.2.3	Justification.....	138
6.2.4	Expected outcomes.....	139
References .....		141
VITA .....		151

## LIST OF TABLES

Table 3.1 Summary of 25 Example Subcellular Locations Extracted by GOcats .....	56
Table 3.2 Agreement Summary between Corresponding GOcats and UniProt CV Subgraphs.....	59
Table 3.3 Agreement Summary between Corresponding GOcats and Map2Slim Subgraphs .....	60
Table 3.4 Summary of 20 Subcellular Locations Used in the HPA Raw Experimental Data Extracted by GOcats.....	62
Table 3.5 Generic Location Categories Used to Resolve Potential Scoping Inconsistencies in HPA Raw Data .....	64
Table 3.6 Summary of Gene Location Category Agreement between Manually-curated HPA Raw Data and GOcats/Map2Slim Categorized HPA-derived Annotations .....	65
Table 4.1 Frequency of Relations in the Gene Ontology and Suggested Semantic Correspondence Classes to Reduce Ambiguity.† .....	99
Table 4.2 Prevalence of Potential <i>has_part</i> Relation Mapping Errors in GO. ....	100
Table 4.3 Summary of GO term Mapping Errors Resulting from Misevaluation of Relations with Respect to Semantic Scoping .....	101
Table 4.4 Uniquely Enriched Terms between GOcats Paths and Traditional Paths from the Breast Cancer Dataset Analysis .....	102
Table 4.5 Binomial Test Results for GOcats Verses Traditional Enrichment for Equine Cartilage Development Time Point Comparisons.....	104
Table 4.6 Neighbor Versus Extreme Time Point Comparison of Enriched Terms in Equine Cartilage Development Enrichment Analyses .....	105
Table 4.7 Comparison of Equine fetus tissue samples .....	106
Table 5.1 Enrichment Results of ANL_45/ANL_60 Pairwise Time-series Comparison for Positively and Negatively Expressed Transcripts, Nested to Show Enrichment of Parent and Child GO Terms.....	120
Table 5.2 Enriched annotations among genes with higher mutational frequency in the KLCG cohort versus the TCGA cohort.....	123

## LIST OF FIGURES

Figure 2.1 UML Diagrams Describing the GOcats Implementation .....	36
Figure 3.1 Network of 25 Categories Whose Subgraphs Account for 89% of the GO Cellular Component Sub-ontology.....	66
Figure 3.2 Network of All Categories from Figure 3.1 Except for Macromolecular Complex.....	67
Figure 3.3 Network of 20 Categories Used in the Human Protein Atlas Subcellular Localization Immunohistochemistry Raw Data.....	68
Figure 3.4 (continued) Visualizing the degree of overlap between the category subgraphs created by GOcats, Map2Slim, and the UniProt CV.....	70
Figure 3.5. Comparison of UniProt-Ensembl knowledgebase annotation data mining extraction performance by GOcats, Map2Slim, and UniProt CV.....	71
Figure 3.6 Comparison of HPA knowledgebase derived annotations to HPA experimental data.....	73
Figure 4.1 GOcats Data Flow Diagram for Creating Categories of GO .....	107
Figure 4.2 The <i>has_part</i> Relation Creates Incongruent Paths with Respect to Semantic Scoping. ....	108
Figure 4.3 Comparison of Adjusted p-values for Significantly-enriched Annotations Using GOcats Paths vs Excluding <i>has_part</i> Edges.....	109
Figure 5.1 Protein-protein interaction network produced after one iteration of additional nodes in STRING from a query of <i>SCN5A</i> , <i>CACNAIS</i> , <i>CACNAIG</i> , <i>KCNMB4</i> , <i>SHISA9</i> , <i>GABRA2</i> , <i>KCND2</i> , <i>ABCA2</i> , <i>CUBN</i> , <i>CACNAII</i> , <i>LRRC38</i> , <i>HCN2</i> , <i>ATPIA3</i> , and <i>CACNG4</i> .....	128
Figure 5.2 Protein-protein interaction network produced after one iteration of additional nodes in STRING from a query of <i>APLP1</i> , <i>ARRB1</i> , <i>NEDD4</i> , and <i>GNAS</i> .....	129
Figure 6.1 Distribution of Word Frequency Versus Word Rank in the Gene Ontology.	140

## LIST OF ADDITIONAL FILES

Supplemental Figures 3.1 A-V Visualizing the degree of overlap between the category subgraphs created by GOcats, Map2Slim, and the UniProt CV ..... [ZIP 723 KB]

Supplemental Table 3.2. List of GO terms mapped by Map2Slim to the term plasma membrane that were not mapped to this location by GOcats ..... [XLSX 13 KB]

Supplemental Table 4.1 Adjusted p-values between omitted has\_part and GOcats part\_of\_some edges for terms enriched in breast cancer data ..... [XLSX 24 KB]

## CHAPTER 1. INTRODUCTION

### 1.1 Ontologies and Their Role in Modern Scientific Research

The word ‘ontology’ is most often associated with its definition from the field of metaphysics: the study of the fundamental nature of being. However, in the field of data science, its second definition provides an arguably more practical—but perhaps less profound—concept: a set of terms within a subject area or domain that defines their properties and the relationships between them. It is this second definition that is referred to in this work, which should hopefully clear up any possible confusion as to why we sometimes refer to multiple ‘ontologies.’

Ontologies may vary greatly on their content, usage, and structure, depending on which field they are designed to serve. However, whether an ontology is designed for describing the structure and procedures of a corporation or the molecular processes within a cell, a few core components are required in some form. These include 1) classes - a basic definition for a collection of objects or individual entities that may be defined extensionally or intensionally; 2) attributes - descriptions, supplementary definitions, or other qualifying information that describe aid in the further description of the class; and 3) relations - descriptions of how one class is related to another within the scope of the ontology. A fourth component, individuals, may also be present, which refer directly to a tangible, real-world instance of a class. For example, the hypothetical class “Chevrolet Malibu automobile” would not be an individual, but an entry specifically referring to Eugene Hinderer’s Chevrolet Malibu by some vehicle identification number attribute would qualify as an individual.

Due to the nature of these components, the most convenient and useful representation of ontological data is a network or graph where classes are represented as nodes, connected by relations that are represented as edges. Depending on the ontology and application, individuals may also be present as nodes, linked to other nodes, or to classes by different relations. Most commonly, classes are related to one another in either a subsumptive relation describing categorical membership, e.g. Chevrolet Malibu *is\_a* automobile, or by a compositional (mereological) relation describing a part of a whole, e.g. wheel *part\_of* automobile. We describe these two types of relationships as “categorization relevant” because they are both useful for grouping classes into collections at ever increasing scope. However, depending on the needs and uses for the ontology in question, additional relations unrelated to subsumptive or mereological membership may also be present describing more complex concepts such as the timing of events, or actions that one class may perform on another class. The complexity of some ontologies necessitates cross referencing to other ontologies that for example, describe how relations relate to one another (1).

Ontologies in the biomedical field include the Systematized Nomenclature of Medicine, Clinical Terms (2)—an ontology standardizing clinical terminology for the storage and retrieval of electronic health data; Chemical Entities of Biological Interest—an ontology describing small chemical compounds relevant to biologist; and the Gene Ontology (GO)—an ontology for describing the cellular locations, molecular functions, and biological processes of genes and gene products (3). These ontologies are indispensable tools for systematically annotating genes, gene products, and other biochemical entities using a consistent set of annotation terms. They are used to document

new knowledge gleaned from nearly every facet of biological and biomedical research today, from classic biochemical experiments elucidating specific molecular players in disease processes to omics-level experiments, providing systemic information on tissue-specific gene regulation. They are created, maintained, and extended by experts with the goal of providing a unified annotation scheme that is readable by humans and machines (4).

GO and other controlled vocabulary databases like the Unified Medical Language System (5,6) saw an explosion in development in the mid-1990s and early 2000s, coinciding with the increase in high-throughput experimentation and “big data” projects like the Human Genome Project. Their intended purpose is to standardize the functional descriptions of biological entities so that these functions can be referenced via annotations across large databases unambiguously, consistently, and with increased automation. However, ontology annotations are also utilized alongside automated pipelines for analyzing protein-protein interaction networks, especially to form predictions of unknown protein function based on these networks (7,8); for gene annotation enrichment analyses that identify conceptual differences between gene sets; and for the creation of predictive disease models in the scope of systems biochemistry (9).

With the advent of transcriptomics technologies, high-throughput investigation of the functional impact of gene expression in biological and disease processes in the form of gene set enrichment analyses represents one important use of GO (10). Many different tools such as Categorizer (11), GOATOOLS (12), and Map2Slim (13) exist to utilize GO annotations in enrichment analyses. These tools solve an essential task of “mapping” specific GO terms to more general GO terms by traversing appropriate edges in the GO

graph structure. However, all current methods fail to utilize all the semantic information available in this ontology, due to inconvenient features in the anatomy of GO.

## 1.2 The Gene Ontology (GO)

### 1.2.1 Overview

The Gene Ontology (GO) (3) is the most common biological ontology used to represent information and knowledge distilled from most biological and biomedical research data generated today, from classic “wet” bench experiments to high-throughput analytical platforms, especially omics technologies. Classes within GO are referred to as *terms*, and each term has several attributes, including a *definition*, which aids in the intensional definition of the term. Each term in GO is also assigned a unique alphanumeric code, which is used to annotate genes and gene products in many other databases, including UniProt (14) and Ensembl (15). Term definitions help researchers determine which term is most necessary for annotating genes in these kinds of databases. Conversely, researchers discovering novel functions or processes performed by a gene or gene product may submit a new term and definition to the GO consortium, which can aid in the expansion and placement of the new term within the ontology.

GO is divided into three sub-ontologies: Cellular Component (CC), Molecular Function (MF), and Biological Process (BP). A graph embodies each sub-ontology, where individual GO terms are nodes connected by directional edges (i.e. relation). For example, the term “connective tissue development” (GO:0061448) is connected by a directional *is\_a* relation edge to the term “tissue development” (GO:0009888). In this graph context, the *is\_a* relation defines the term “tissue development” as a parent of the term “connective

tissue development”. Likewise, “tissue development” (GO:0009888) is\_a “anatomical structure development” (GO:0048856), which in turn is\_a “developmental process” (GO:0032502). From a GO term mapping perspective, “connective tissue development” (GO:0061448) is\_a “developmental process” (GO:0032502). Similar pedigree-like or genealogical terminology is used to describe the relations between terms; here, we would refer to “connective tissue development” as the child of “tissue development,” and we could also speak of “ancestors” or “descendants” of these terms by following directional relations up or down the hierarchy of the graph. There are eleven types of relations used in the core version of GO; however, *is\_a* is the most ubiquitous. The three GO sub-ontologies are “*is\_a* disjoint” meaning that there are no *is\_a* relation edges connecting any node among the three sub-ontologies. However, other relations, such as “regulates,” connect nodes of separate sub-ontologies. Relations of interest to this study are *part\_of* and *has\_part*. These are like *is\_a* in that they describe scope, i.e. relative generality or encompassment, but are separate in that *is\_a* represents true sub-classing of terminology while *part\_of* and *has\_part* describe mereological correspondence. Therefore, we consider scoping relations to be comprised of *is\_a*, *part\_of*, and *has\_part*, and mereological relations to be comprised of *part\_of* and *has\_part*.

### 1.2.2 Data structure

The data structure of GO follows the guidelines set forth by the OBO Foundry (4), meaning that it is available in the OBO format or in the Web Ontology Language (OWL) format (GO is available in both) and that it adheres to OBO’s principals, which are: 1) the ontology is open-access, 2) it is expressed in a common formal language, 3) it possesses a

unique identifier space within OBO (hence why GO terms begin with the identifier “GO:”), 4) versions of each ontology are clearly specified, 5) the content of each ontology is cleanly delineated, 6) contextual definitions are provided for all terms, 7) relations are unambiguous, 8) the ontology is well-documented, 9) there is a plurality of independent users, and 10) the ontology’s development is collaborative (16). As many of these points are subjective, developers within the OBO Foundry review candidate ontologies for inclusion as members.

There are three versions of the GO database: *go-basic* which is filtered to only include *is\_a* and *part\_of* relations; *go* or *go-core* which contains additional relations that may span sub-ontologies and which point both toward and away from the top of the ontology; and *go-plus* contains yet more relations in addition to cross-references to entries in external databases like the Chemical Entities of Biological Interest ontology (17). The first and second versions are available in the Open Biomedical Ontology (OBO) flat text file formatting, while the third is available only in the Web Ontology Language (OWL) RDF/XML format. In this project, we utilized the OBO flat file format. This format is comprised of a header, which contains information about the version of the ontology, as well as other metadata, and *stanzas* which define terms and relations. Each term stanza contains the GO identifier code and various attributes such as the term name, definition, and references to direct parent terms. Below is an example of a term stanza taken directly from the GO OBO file:

```
[Term]
id: GO:0000001
name: mitochondrion inheritance
namespace: biological_process
```

```

def: "The distribution of mitochondria, including the
mitochondrial genome, into daughter cells after mitosis or
meiosis, mediated by interactions between mitochondria and
the cytoskeleton." [GOC:mcc, PMID:10873824, PMID:11389764]
synonym: "mitochondrial inheritance" EXACT []
is_a: GO:0048308 ! organelle inheritance
is_a: GO:0048311 ! mitochondrion distribution

```

Here we can see the relational link between “mitochondrion inheritance” and its two parent terms, “organelle inheritance” and “mitochondrion distribution,” each associated by an *is\_a* relation.

Stanzas defining relations are labeled as *Typedef* and contain cross references to entries in a higher level Basic Formal Ontology (BFO) (18) to aid in potential disambiguation of relationship meanings. The following is an example of a *Typedef* stanza in GO:

```

[Typedef]
id: has_part
name: has part
namespace: external
xref: BFO:0000051
is_transitive: true

```

Here we can see that the *has\_part* relation is defined externally in the BFO and is transitive.

While the OBO flat file format is complete enough for parsing, representative of the graph structure of GO, and generally human-readable, the OWL format version is more structured and amenable for use with common ontology editors like Protégé (19).

### 1.2.3 Gene Ontology Annotations

Terms defined within ontologies such as GO are referred to as annotations when they are associated with an entity in another database. At this level, the ontological terms/classes themselves can be thought of as an attribute of the entity in the other database. Since GO terms are constantly being updated to match the most current scientific findings and definitions, GO annotations are intended to represent the current snapshot of biological knowledge (20).

GO annotations are assigned by database curators based on one of several evidence codes, which fall into six categories: 1) experimentally based, which include assays, expression patterns, and physical interactions determined by direct experiments described in literature; 2) phylogenetically-inferred, in which gene functions are inferred based on gain and loss of functions of phylogenetically-related genes; 3) computational analysis, which includes functional inferences based on sequence or structural similarity determined by *in silico* techniques; 4) author statements, which include direct statements that the authors made regarding gene functions in the literature; 5) curator statements, which involves functional assignments based on the judgment of the database curator assigning the annotation; and 6) electronic annotation evidence, which are not, or not yet, manually reviewed (21). This sixth category has annotations assigned based on three automated processes. The first is an assignment of annotations based on associations that each GO term has with a sequence signatures for groups of homologous proteins. Interpro2GO (22) and PANTHER (23) are common tools used for this purpose. A second method involves the conversion of terms within the UniProt controlled vocabulary (14,24) into associated

GO terms. Finally, a third method involves inferring annotations from orthologous genes available through the Ensembl database (15).

GO annotations are provided in a standardized file format called a gene annotation file (GAF). The current version of this format, GAF 2.1, is provided as a tab-delimited table with a number of required or optional columns, i.e. fields. Critical required fields include: *database*, indicating which database the gene or gene product originates from (e.g. UniProt); *database object id*, which is the unique database identifier code for the gene or gene product; *database object symbol*, which is the gene symbol associated with the entry (e.g. PHO3); *GO ID*, which is the GO term annotation associated with the entry (a 1:1 mapping of each GO term to each gene or gene product is maintained in the dataset, so entries are repeated for every GO annotation in the file); and *evidence code*, indicating which of the previously-described evidence categories is responsible for the gene annotation. GAFs are created and maintained per species and are provided by the Gene Ontology Consortium. The human GOA is available through the European Molecular Biology Laboratories-European Bioinformatics Institute (EMBL-EBI) FTP server (25).

### 1.3 Annotation Enrichment and the Importance of Ontological Inference

Annotation enrichment analysis is one of the most common uses for gene annotations, based on associated biomedical ontology terms. In the most basic sense, annotation enrichment is an analysis of which biological concepts are statistically over-represented in a gene set from an experimental condition versus a gene set from a control condition. Commonly, this enrichment is performed on gene expression results generated from high-throughput transcriptomic analyses that demonstrate quantifiable changes in gene expression between experimental and control systems. However, technically any

method that can distinguish a *foreground* subset of genes from the *universe* (the whole set of genes in the experiment) in a control condition and at least one experimental condition is amenable to enrichment. Examples include results from a DeSEQ2 (26) analysis of transcript expression levels, where foreground gene sets may be selected based on significantly increased or decreased transcript expression levels in an experimental condition, such as a disease model versus control, and results from a MutSig (27) analysis of mutational profiles in cancer patients' DNA sequencing results, where foreground genes are selected based on which genes are mutated more often than what would be expected by random chance.

Annotation enrichment calculates the likelihood that at least  $x$  number of genes out of  $n$  number of total genes in the foreground gene set share the same annotation (GO term) by random chance, considering the distribution of that annotation among the genes in the universe. This likelihood is given by a p-value determined by a variety of statistical methods (28). Here, it is pertinent to emphasize that annotation enrichment is a discovery-based analysis that is designed to infer information that is inherent to the data in question and is not hypothesis-driven by any ground truth. In other words, the null-hypothesis used to derive p-values from the statistical tests is based on how the test statistic fits an expected mathematical distribution; it does not take any biochemically-relevant parameters into account, other than the method used to produce the foreground gene set.

While it is possible to perform annotation enrichment while only considering the direct GO terms annotated to each gene, the graph nature of GO allows for inferences to be made such that ancestor terms can be included as annotations for genes as well. As mentioned, some relations that form the edges of this graph are not relevant for semantic

categorization. Therefore, which paths are safe to follow in these inference paths is a serious point of consideration. Still, the inclusion of ontological ancestors in annotation enrichment analyses helps improve the interpretation of the results by providing more generalized information to help summarize the enrichment and to help account for the fact that genes are often annotated at varying levels of granularity, depending on the methods used in determining their annotation.

Relevant to this work, CategoryCompare (29) is an analytical tool, developed by Dr. Robert Flight, which can calculate annotation enrichment of annotations as well as their ontological ancestor terms, from a provided gene set and their annotations. This tool uses a hypergeometric test to determine the significance of each enriched annotation and provides an adjusted p-value using a Benjamini-Hochberg correction for multiple hypothesis testing (30).

#### 1.4 Difficulty in Representing Biological Concepts Derived from Omics-Level

##### Research

Differential abundance analyses for a range of omics-level technologies, especially transcriptomics technologies can yield large lists of differential genes, gene-products, or gene variants. From annotation enrichment analysis, many different enriched GO annotation terms may be associated with these differential gene(-product) lists, making it difficult to interpret without manually sorting into appropriate descriptive categories (11). It is similarly non-trivial to give a broad overview of a gene set or make queries for genes with annotations of a biological concept. For example, a recent effort to create a protein-protein interaction network analysis database resorted to manually building a hierarchical localization tree from GO cellular compartment terms due to the “incongruity in the

resolution of localization data” in various source databases and the fact that no published method existed at that time for the automated organization of such terms (7). If subgraphs of GO could be programmatically extracted to represent such concepts, a category-defining general term could be easily associated with all its ontological child terms.

Meanwhile, high-throughput transcriptomic and proteomic characterization efforts like those carried out by the Human Protein Atlas (HPA) now provide sophisticated pipelines for resolving expression profiles at organ, tissue, cellular and subcellular levels by integrating quantitative transcriptomics with microarray-based immunohistochemistry (31). Such efforts create a huge amount of omics-level experimental data that is cross-validated and distilled into systems-level annotations linking genes, proteins, biochemical pathways, and disease phenotypes across our knowledgebases. However, annotations provided by such efforts may vary in terms of granularity, annotation sets used, or ontologies used. Therefore, (semi-)automated and unbiased methods for categorizing semantically-similar and biologically-related annotations are needed for integrating information from heterogeneous sources—even if the annotation terms themselves are standardized—to facilitate effective downstream systems-level analyses and integrated network-based modeling.

## 1.5 Term Categorization Approaches

Issues of term organization and term filtering have led to the development of GO slims—manually trimmed versions of the gene ontology containing only generalized terms (32), which represent concepts within GO. Other software, like Categorizer (11), can organize the rest of GO into representative categories using semantic similarity measurements between GO terms. GO slims may be used in conjunction with mapping

tools, such as OWLTools' Map2Slim (M2S) (13) or GOATools (12), to map fine-grained annotations within Gene Annotation Files (GAFs) to the appropriate generalized term(s) within the GO slim or within a list of GO terms of interest. While web-based tools such as QuickGO exist to help compile lists of GO terms (33), using M2S either relies completely on the structure of existing GO slims or requires input or selection of individual GO identifiers for added customization, and necessitates the use of other tools for mapping. UniProt has also developed a manually-created mapping of GO to a hierarchy of biologically-relevant concepts (24). However, it is smaller and less maintained than GO slims, and is intended for use only within UniProt's native data structure.

## 1.6 Semantic Similarity in the Context of Broad Term Categorization

In addition to utilizing the inherent hierarchical organization of GO to categorize terms, other metrics may be used for categorization. For instance, semantic similarity can be combined along with the GO structure to calculate a statistical value indicating whether a term should belong to a predefined group or category (11,34–37). One rationale for this type of approach is that the topological distance between two terms in the ontology graph is not necessarily proportional to the semantic closeness in meaning between those terms, and semantic similarity reconciles potential inconsistencies between semantic closeness and graph distance. Additionally, some nodes have multiple parents, where one parent is more closely related to the child than the others (11). Semantic similarity can help determine which parent is semantically more closely related to the term in question. While these issues are valid, we maintain that in the context of aggregating fine-grained terms into general categories, these considerations are not necessary. First, fluctuations in semantic distances between individual terms are not an issue once terms are binned into

categories: all binned terms will be reduced to a single step away from the category-defining node. Second, the problem of choosing the most appropriate parent term for a GO term only causes problems when selecting a representative node for a category; however, since most paths eventually converge onto a common ancestor, any significantly diverging paths would have its meaning captured by rooting multiple categories to a single term, cleanly sidestepping the issue.

### 1.7 Maintenance of Ontologies

Despite maintenance and standard policies for adding terms, ontological organization is still subject to human error and disagreement, necessitating quality assurance and revision, especially as ontologies evolve or merge. A recent review of current methods for biomedical ontology mapping highlights the importance in developing semi-automatic methods (38,39) to aid in ontology evolution efforts and reiterates the aforementioned concept of semantic correspondence in terms of scoping between terms (40). Methods incorporating such correspondences have been published elsewhere, but these deal with issues of ontology evolution and merging, and not with categorizing terms into user-defined subsets (41,42). Ontology merging also continues to be an active area of development for integrating functional, locational, and phenotypic information. To aid in this, another review points out the importance of integrating phenotypic information across various levels of organismal complexity, from the cellular level to the organ system level (9). Thus, organizing location-relevant ontology terms into discrete categories is an important step toward this end.

## 1.8 Path Traversal Issues in GO

Ontological graphs are typically designed as directed graphs, meaning that every edge has directionality, or directed acyclic graphs (DAGs), meaning that no path exists that leads back to a node already visited if one were to traverse the graph stepwise. This allows the graph to form a complex semantic model of biology containing both general concepts and more-specific (fine-grained) concepts. The “parent-child” relation hierarchy allows biological entities to be annotated at any level of specificity (granularity) with a single term code, as fine-grained terms intrinsically capture the meaning of every one of its parent and ancestor terms through the linking of relation-defining *is\_a* edges in the graph. However, it is deceptively non-trivial to reverse the logic and organize similar fine-grained terms into general categories—such as those describing whole organelles or concepts like “DNA repair” and “kinase activity”—without significant manual intervention. This is due, in part, to the lack of explicit scoping, scaling, and other semantic correspondence classifiers in relations. Therefore, it is not readily clear how to classify terms connected by non-*is\_a* relation edges. Although edges are directional, the semantic correspondence between terms connected by a scoping relation is computationally ambiguous, e.g. assessing whether term 1 is more/less general or equal in semantic scope with respect to term 2 is currently not possible without explicitly defining rules for such situations.

Ambiguity in assessing which term is more general in a pair of terms connected by a relation edge is confounded by the fact that edges describing mereological relations, such as *part\_of* and *has\_part*, are not strictly and universally inverse of one another. For instance, while every “nucleus” is *part\_of* “cell,” not every “cell” *has\_part* “nucleus.” Similarly, while every “nucleus” *has\_part* “chromosome,” not every “chromosome” is

*part\_of* “nucleus” under all biological situations. Therefore, mereological edges are not necessarily reciprocal. Ontological logic rules, called axioms, ensure that this logic is maintained in the graph representation by allowing edges of the appropriate type to connect terms only if the inferred relation is universal (43). GO maintains its own set of axioms regarding the relations it contains (44). This axiomatic representation is crucial to avoid making incorrect logical inferences regarding universality but does nothing to facilitate categorization of terms into parent concepts, especially since some mereological edges point away from the root of the ontology toward a narrower scope. If these edges are followed, terms of more broad scope may be grouped into terms of more narrow scope, or worse, cycles may emerge which would abolish term hierarchy and make both categorization and semantic inference impossible. To circumvent this problem, some ontologies release versions that do not contain these types of edges. For GO, this is accomplished by go-basic. However, information is lost when these edges are removed from the graph. When attempting to organize fine-grained terms into common concepts using the hierarchical structure, this information loss can be significant because many specific-to-generic term mappings can utilize the same edge in many paths.

## 1.9 Axiomatic Versus Semantic Scoping Interpretation of Mereological Relations in GO

Ensuring mereological universality in relation associations using current axioms is important within the purview of ontology development. However, for those interested in organizing datasets of gene annotations into relevant concepts for better interpretation, such is the case in annotation enrichment, it is important to utilize the full extent of the information within an ontology. Current axiomatic representation of mereological relations

requires the use of ontology versions which lack certain relations (32), resulting in a loss of retrievable information. If *has\_part* edges, which point toward terms of narrower scope, were inverted to resemble *part\_of* edges, ensuring that all edges point toward terms of a broader scope, terms could be effectively categorized with respect to semantic scope using the native graph hierarchy without losing any information in the process. However, this isn't logically possible because of issues dealing with universality.

Issues regarding ambiguity and other shortcomings of ontological relations, especially in GO, have been reported as far back as 2005 (45), which contributed to the development of the Relations Ontology (1). Such studies point to possible solutions to the correct interpretation of the problematic *has\_part* relation. One such case is to include a relation called *integral\_part\_of* to provide a reversible *part\_of* relation for cases where A *part\_of* B and B *has\_part* A maintains a universal sense. However, this case still does not address how non-universal instances should be dealt with. Furthermore, despite the effort in building a full ontology for relations, the OBO Foundry still does not require nor even officially recommend that the Relations Ontology be integrated with the other ontologies in the OBO due to the fact that other OWL ontologies use instance-level relations, while OBO ontologies use type-level relations (46). Therefore, there is still no standard conventional method for dealing with relations like *has\_part* other than ignoring them altogether.

We acknowledge the importance of existing axioms, which prohibit reversing mereological edges in ontologies under the context of drawing *direct* semantic inferences. However, we maintain that in the context of detecting enriched broad concepts based on “summarizing” annotated fine-grained terms contained within differential annotation

datasets, it is appropriate to evaluate mereological relations from a scoping perspective, which requires that all mereological edges point to their whole. This conundrum preventing the comprehensive categorization of GO terms can be dealt with by adding a single new relation to the ontology: *part\_of\_some*. Semantically, this relation deals with both the issue of universality and with the issue of the direction of granularity.

## CHAPTER 2. MATERIALS AND METHODS

### 2.1 The Gene Ontology Categorization Suite (GOcats)

#### 2.1.1 Methodological Overview and Design Rationale for GOcats

We designed the Gene Ontology Categorization Suite (GOcats) with a biologist user in mind, who may not be aware of the dangers associated with using different versions of GO for organizing terms with tools like M2S or how to circumvent potential pitfalls. For instance, although the M2S documentation (47) states, "We recommend the go-basic version of the ontology be used, which contains: subClassOf (is a), part of, regulates (+ positively and negatively regulates)" and, "You can also use the full version of GO and filter those relationships you do not want to consider," a non-bioinformatician may not be aware of how to filter out relationships from GO in a way that is safe to use. More pertinently, the user may wish to use a fuller extent of the information contained in the ontology when organizing their terms but be unable to do so safely on their own. Currently, GOcats version 1.1.4 can handle go-core's *is\_a*, *part\_of*, and *has\_part* relations, with the *has\_part* reinterpreted to retain proper scoping semantics, as detailed below and elsewhere (48). As the development of GOcats progresses, we plan on handling the organization of terms connected by additional relations such as *negatively\_regulates* or *positively\_regulates*.

GOcats uses the go-core version of the GO database, which contains relations that connect the separate ontologies and may point away from the root of the ontology. GOcats can either exclude non-scoping relations or invert *has\_part* directionality into a

*part\_of\_some* interpretation, maintaining the acyclicity of the graph. Therefore, it can represent go-core as a DAG.

GOcats is a Python package written in major version 3 of the Python program language (49) and available on GitHub and the Python Package Index (50,51). It uses a Visitor design pattern implementation (52) to parse the go-core Ontology database file (5). Searching with user-specified sets of keywords for each category, GOcats extracts subgraphs of the GO DAG and identifies a representative node for each category in question and whose child nodes are detailed features of the components. Details are provided in Chapter 2.1.3.

To address issues regarding scoping ambiguity among mereological relations, we assigned properties indicating which term was broader in scope and which term was narrower in scope to each edge object created from each of the scope-relevant relations in GO. For example, in the node pair connected by a *part\_of* or *is\_a* edge, node 1 is narrower in scope than node 2. Conversely, node 1 is broader in scope than node 2 when connected by a *has\_part* edge. This edge is therefore reinterpreted by GOcats as *part\_of\_some*. This reinterpretation is not meant to imply exclusivity in composition between the meronym and the holonym. It simply stands as a distinction between “part of all” which is what the current *part\_of* relationship implies, and “part of some,” or to be more verbose “instance a is part of instance b in at least one known biological example.” We have described additional explanations and rationale for this re-interpretation elsewhere and demonstrate improvement in annotation enrichment analyses across GO Cellular Component, Molecular Function and Biological Process sub-ontologies, when this re-interpretation is used (see Chapter 4).

While the default scoping relations in GOcats are *is\_a*, *part\_of*, and *has\_part*, the user has the option to define the scoping relation set. For instance, one can create go-basic-like subgraphs from a go-core version ontology by limiting to only those relations contained in go-basic. For convenience, we have added a command line option, “go-basic-scoping,” which allows only nodes with *is\_a* and *part\_of* relations to be extracted from the graph. Detailed API documentation and user-friendly tutorials are available online (53).

For term mapping purposes, Python dictionaries are created, which map GO terms to their corresponding category or categories. For inter-subgraph analysis, another Python dictionary is created, which maps each category to a list of all its graph members. By default, fine-grained terms map to the closest category root node, when multiple category root node mappings are possible. In other words, a fine-grain term will not map to a category root-node that define a subgraph that is a superset of a category with a root-node nearer to the term. For example, a member of the “nucleolus” subgraph would map only to “nucleolus,” and not to both “nucleolus” and “nucleus”. However, the user also has the option to override this functionality if desired with a simple “--map-supersets” command line option. Furthermore, we’ve included the option for users to directly input GO terms as category representatives, should they not wish to use keywords to define subgraph categories. Also, the user can use a combination of categories defined by either keyword and/or representative GO term. This is helpful for users who have already compiled lists of GO terms by hand for use with other tools.

### 2.1.2 GOcats Implementation Overview

As illustrated in the UML diagram in Figure 2.1A, the GOcats package is implemented using several modules that have clear dependencies starting from a command line interface (CLI) in `gocats.py`, which depend on most of the other modules including `ontologyparser.py`, `godag.py`, `subdag.py` and `tools.py`. GOcats uses 10 classes implemented across `ontologyparser.py`, `godag.py`, `subdag.py`, and `dag.py` modules to extract and internally represent the GO database. `GoParser`, which inherits from the base `OboParser` class (Figure 2.1B), utilizes a Visitor design pattern and regular expressions to parse the flat GO database obo file and instantiate the objects necessary to represent the GO DAG structure. These instantiated objects include (Figure 2.1C): 1) the `GoGraph` container object for the parts of the graph, whose class inherits from a more generic `OboGraph` class, containing functions for adding, removing, and modifying nodes and edges; 2) `GoGraphNode` objects for representing each term parsed from the ontology, whose class inherits from `AbstractNode` class; 3) `AbstractEdge` objects for representing each instance of a relation parsed from the ontology; and 4) `DirectionalRelationship` objects, whose class inherit from the more generic `AbstractRelationship` class for representing each type of directional relation encountered in the ontology (for GO, all relations are directional, and this distinction is made only in anticipation for future extensions to handle other ontologies with non-directional relationships).

`AbstractEdge` and `AbstractNode` objects contain references to one another, which simplifies the process of iterating through ancestor and descendant nodes and allows for functions such as `AbstractEdge.connect_nodes`, which requires that the edge object update

the node object's `child_node_set` and `parent_node_set`. In this context, `AbstractNode` is a true abstract base class, while `AbstractEdge` started out as an abstract base class but eventually became a concrete class during development. However, we see the possibility of `AbstractEdge` becoming a base class in the future.

Ancestors and descendants of a node are implemented as sets, which are lazily created using a Python property decorator (i.e. Python's preferred "getter" syntax). At the first access of these sets through the ancestor or descendent property, the set is calculated with a recursive algorithm, stored for future use, and returned for immediate access. Subsequent accesses simply return the stored set. If the set of edges within a node change, the ancestor and descendent node sets will be recalculated on their next access. This implementation prevents pre-calculation of these sets when they are not used, while enabling their reuse within efficient graph analysis methods.

`AbstractEdge` also contains a reference to a `DirectionalRelationship` object, which is critical for graph traversal. This is because the `DirectionalRelationship` object contains the true directionality of the mereological correspondence between the categorization relevant relations (*is\_a*, *part\_of*, and *has\_part*). In other words, it is within this object that we define in which direction the edge should be traversed when categorizing terms. Currently, these rules are hard-coded within `GoParser`'s `relationship_mapping` dictionary.

The `gocats.py` module (Figure 2.1A) implements the command line interface and is responsible for handling the command line arguments, using the provided keywords and specified arguments like namespace filters (e.g. Cellular Component, Molecular Function, and Biological Process) to instantiate a `GoParser` object, a `GoGraph` object, and a `SubGraph` object for each set of provided keywords or representative GO terms. After creation of the

GoGraph internal representation, each category subgraph is created by first instantiating the SubGraph object and calling the `from_filtered_graph` function, which filters to those nodes from the GoGraph containing the keywords in their names and definition. Note that the SubGraph object and GoGraph object both inherit from OboGraph, and that the SubGraph object contains a reference to GoGraph object (supergraph data member) of which it is a subgraph. This design was implemented to avoid accidental alterations of the GoGraph object when altering the contents of the subgraph, and to allow for specialization of functions within SubGraph without needing to use unique names such as `add_subgraph_node()` when `add_node()` would suffice. GoGraphNode objects within the subgraph are wrapped by SubGraphNode objects, which are directly used by the SubGraph object, but retain all original properties such as name, definition, and sets of edge object references, otherwise insidious changes could occur to the GoGraph object when updating the SubGraph object. The SubGraph object also contains a CategoryNode object, which wraps the category representative GoGraphNode object(s) for the subgraph category.

### 2.1.3 GOcats Specific Implementation Details

User-provided keyword sets are used by GOcats to query GO terms' name and definition fields to create an initial seeding of the subgraph with terms that contain at least one keyword. This seeding is a list of nodes from the whole go-core graph (supergraph) that pass the query. Node synonyms were not used, due to there being four types of synonyms in GO: exact, narrow, broad, and related. Also, many nodes within GO do not have synonyms, which may create an unequal utilization of nodes if synonyms were queried. However, in the future, synonym utilization for seeding purposes may be revisited.

```
FOR node in supergraph.nodes
```

```
    IF keyword from keyword_list in node.name /  
        or node.definition
```

```
        APPEND node to subgraph.seeding_list
```

Using the graph structure of GO, edges between these seed nodes are faithfully recreated except where edges link to a node that does not exist in the set of newly seeded GO terms. During this process, edges of appropriate scoping relations are used to create children and parent node sets for each node.

```
FOR edge in supergraph.edges
```

```
    IF edge.parent_node in subgraph.nodes AND /  
        edge.child_node in subgraph.nodes AND /  
        edge.relation is TYPE: SCOPING
```

```
        APPEND edge to subgraph.edges
```

```
    ELSE
```

```
        PASS
```

```
FOR subnode in subgraph.nodes
```

```
    subnode.child_node_set = {child_node for child_node in /  
        supergraph.id_index[subnode.id].child_node_set if /  
        child_node.id in subgraph.id_index}
```

```
    subnode.parent_node_set = {parent_node for parent_node /  
        in supergraph.id_index[subnode.id].parent_node_set if /  
        parent_node.id in subgraph.id_index}
```

GOcats then selects a category representative node to represent the subgraph. To do this, a list of candidate representative nodes is compiled from non-leaf nodes, i.e. root-nodes in the subgraph which have at least one keyword in the term name. A single category

representative root-node is selected by recursively counting the number of children each candidate term has and choosing the term with the most children.

```
FOR subnode in subgraph
    IF subnode.child_node_set != None AND ANY keyword in /
        subnode.name
        candidate_list.append(subnode)
    ELSE
        PASS
representative_node = MAX(LEN(node.descendants) FOR node /
    in candidates)
```

Because it may be possible that highly-specific or uncommon features included in the GO term may not contain a keyword in its name or definition but still may be part of the subgraph in question by the GO graph structure, GOcats re-traces the supergraph to find various node paths that reach the representative node. We have implemented two methods for this subgraph extension: i) comprehensive extension, whereby all supergraph descendants of the representative node are added to the subgraph and ii) conservative extension, whereby the supergraph is checked for intermediate nodes between subgraph leaf nodes and the subgraph representative node that may not have seeded in the initial step.

#### Comprehensive extension:

```
FOR node in supergraph
    IF ANY (ancestor_node in node.ancestors) in subgraph
        subgraph_nodes.append(ancestor_node)
UPDATE subgraph
```

#### Conservative extension:

```

FOR leaf_node in subgraph.leaf_nodes # nodes with no children
    start_node = leaf_node
    end_node = representative_node
    FOR node in super_graph.start_node.ancestors  $\cap$  /
        supergraph.end_node.descendants
        subgraph_nodes.append(node)
UPDATE subgraph

```

The subgraph is finally constrained to the descendants of the representative node in the subgraph. This excludes unrelated terms that were seeded by the keyword search due to serendipitous keyword matching.

#### 2.1.4 Defining and Traversing Categorization-relevant Edges in GO

As mentioned in Chapter 2.1, we equipped GOcats with the ability to deal with the problematic *has\_part* relation by re-evaluating it with the logic of *part\_of\_some*. While the semantic logic explained in that chapter is accurate with regard our intention of that interpretation in the scope of annotation enrichment, it is important to stress here that this reinterpretation is not accomplished by natural language processing (NLP), although we plan on implementing these types of interpretations in the future (see Chapter 6.1.2).

In our current version of GOcats 1.1.4c, handling of directional edge traversal is agnostic of relation semantics. Instead, “direction” is a data member of the *DirectionalEdge* class and determines whether the “forward\_node” or “reverse\_node” object reference is accessed during a traversal event. The “forward” and “reverse” nodes are assigned based on the order in which the edge reference is encountered, while parsing the GO database file and is always referenced in the same way regardless of the intended

edge directionality with respect to the GO hierarchy. Let's use the following stanza as an example.

```
[Term]
id: GO:0000243
name: commitment complex
namespace: cellular_component
def: "A spliceosomal complex that is formed by association of the U1 snRNP with the 5' splice site of an unspliced intron in an RNA transcript." [GOC:krc, ISBN:0879695897, PMID:9150140]
synonym: "mammalian spliceosomal complex E" NARROW [GOC:krc, GOC:mah, ISBN:0879695897, ISBN:0879697393]
synonym: "mammalian spliceosomal E complex" NARROW [GOC:mah]
synonym: "yeast spliceosomal complex CC" NARROW [GOC:krc, GOC:mah, ISBN:0879695897, ISBN:0879697393]
is_a: GO:0005684 ! U2-type spliceosomal complex
relationship: has_part GO:0005685 ! U1 snRNP
```

The “commitment complex” term would be the reverse node of both the *is\_a* relation edge and the *has\_part* relation edge even though these edges point in opposite directions in the GO hierarchy. Within GOcats, there is currently a hard-coded mapping indicating conventional hierarchical directionality (0 for reverse\_node → forward\_node), and inversed directionality (1 for forward\_node → reverse\_node). During traversal, this Boolean is checked within each edge type to determine which node to follow along the path. We constructed this simple hard-coded edge directionality mapping in such a way as to make it straightforward to integrate more sophisticated NLP-enabled evaluation of relations in the future, as the results of such evaluations would need only to update a single mapping dictionary value for each relation to function within the remaining code base.

## 2.2 Pipelines Incorporating GOcats’ Ancestor Paths and Categorizations into Annotation Enrichment Analyses

While GOcats creates augmented ontological graph representations for the purpose of improving annotation enrichment analyses and data visualization, it does not contain built-in annotation enrichment algorithms or methods for accessing visualizations of some

common applications such as protein-protein interaction networks as of version 1.1.4c. Therefore, we have created scalable and reproducible data workflow pipelines using the Snakemake workflow management system (54) to integrate GOcats with CategoryCompare2 (29), an annotation enrichment software tool, as well as with the REST API of STRING (55) for visualizing protein-protein interaction networks for genes associated with enriched annotations.

For annotation enrichment, our goal was to perform enrichment not only on direct gene annotations, but also across all ontological ancestor terms, i.e. those terms that are more general and above the direct terms in the ontological hierarchy. To accomplish this, we implemented a function in GOcats called `build_graph_interpreter` (56), which builds complete lists of ancestor ontology terms for all genes listed in the gene annotation file (GAF) of the organism in question. This function is part of the GOcats API and was designed to quickly build an ontology graph object representation within a Python interpreter. Mappings of gene symbols to their comprehensive list of annotations and ancestor annotations were output into a JSON file format that would later be input into CategoryCompare2 for enrichment within Snakemake workflows. In this way, we were able to utilize GOcats' reinterpretation of relations as described in Chapter 2.1.4 for annotation enrichment applications. This workflow step was coded into a Snakemake rule called *build\_ancestor\_list.smk* and was run as a first step in each workflow that utilized GOcats' ancestor paths. For testing purposes, we also ran enrichments which mimicked traditional path tracing by intentionally omitting the `has_part` relation during this step. To do this, we utilized a command line option in GOcats: "`--allowed_relationships=[is_a,`

part\_of]”, which overrides GOcats’ default path tracing algorithms and creates the GO graph representation using only the relations specified in the provided list (ref 57, SD4)

Each Snakemake workflow is executed by a top-level script, conventionally named Snakefile, which is responsible for loading in user-supplied parameters from a configuration file, as well as the Snakemake rule scripts and workflow scripts that execute lower-level tasks and subroutines. Within each script, syntax for defining the required input and output files dictate the order of operation in which the scripts are run, and this functionality is native to Snakemake (54). The difference between rule and workflow scripts are subtle and non-explicit in terms of base syntax, but in practice, workflow scripts operate at a higher level, and often include rule scripts as subroutines.

We implemented the base-level annotation enrichment tasks within a rule script called *enrichment\_rules.smk*. Rules within this script include: 1) *create\_annotations*, which is responsible for taking the annotation JSON file produced by GOcats within the *build\_ancestor\_list.smk* script and converting it into a format that CategoryCompare2 can use for annotation enrichment; 2) *generate\_gene\_sets*, which parses the user-supplied dataset of genes or gene products and organizes the dataset into distinct feature sets, usually significant genes of interest and universe, but depends on the application; 3) *generate\_feature\_files*, which converts the previously identified features into a format that CategoryCompare2 can use for annotation enrichment; 4) *run\_enrichment*, which is responsible for executing CategoryCompare2’s annotation enrichment algorithms; and 5) *generate\_enrichment\_results*, which performs additional formatting of results, like the addition of GO term descriptions and the addition of associated genes for each enriched term in the results. Excepting some minor formatting steps, like retrieving GO term

descriptions for addition to the resulting enrichment table, this collection of rules contains all the steps necessary for performing a single annotation enrichment analysis.

For complex enrichment analyses, such as the time-series enrichment described in Chapters 4.5.5 and 5.1.1.1, we designed a higher-level enrichment workflow for handling a series of consecutive enrichment analyses. First, all required enrichment analyses are enumerated based on the number of input datasets supplied. This can be customized within the top-level Snakefile script but is set to enumerate based on the number and names of the sheets in the DEseq2 data Excel spreadsheets supplied by our collaborators for our time-series enrichments. For each pairwise, or whole time series enrichment analysis (see Chapter 5.1.1.1), a subdirectory is created, and the “enrichment” section of the top-level configuration file is copied into a new configuration file and placed within each subdirectory. The higher-level script, called *enrichment\_subworkflow.smk*, contains a rule called *single\_enrichment\_workflow* which navigates into each subdirectory and executes the base-level single enrichment script described previously. Because each subdirectory contains a copy of the necessary configuration details, we are able to seamlessly reuse the base-level enrichment rules; the scripts are executed as if they are being run on a single annotation enrichment analysis.

We also utilized Snakemake for comparing the performance of GOcats’ path tracing and traditional ontology path tracing as they relate to enrichment results. Specifically, we compared the resulting adjusted p-values from the time series equine cartilage tissue transcript annotation enrichments when using GOcats ontological ancestor paths to the traditional ancestor path tracing method using a binomial test (see Chapter 4.2.2). As mentioned previously, we used GOcats’ “--allowed\_relationships” command

line option to override the default path tracing algorithms and mimic traditional path tracing methods by including only those relations that traditional path tracing methods use: *is\_a* and *part\_of*. When performing performance comparisons, we added an extra “ancestor\_traversal” parameter in the configuration file and forced the workflow to produce enrichments across all pairwise and whole time-series gene sets for each traversal method. This required that the aforementioned *build\_ancestor\_list.smk* script be run twice, and all previously described enrichment methods to be run for each path traversal type.

For comparing results, we implemented a binomial test within a script named *binomial\_test\_rules.smk*. This script was executed at the end of the top-level *Snakefile* script and was responsible for reading and comparing all enrichment tables produced using GOcats path traversal and traditional path traversal. Every enriched GO term from each enrichment table was mapped to their respective adjusted p-value for each path traversal method. For those terms with an adjusted p-value less than 0.01, the script tested whether the value was lower (more significant) in the GOcats path traversal method. Identical values were ignored, and not added to the total number of comparisons. We performed a binomial test using the *SciPy stats* (58) Python module, comparing the number of times GOcats’ derived enrichments had a lower adjusted p-value than the traditional path traversal algorithm to the number of time the traditional path traversal method’s enrichment p-values were lower, assuming a null hypothesis of 0.5 and using a one-sided test.

## 2.3 Visualizing Protein-Protein Interaction Network Visualizations based on Enrichment Results

To leverage annotation enrichment results within the context of protein-protein interaction networks, we first performed annotation enrichment on datasets taken from

gene mutational frequency analysis of whole-genome sequences among cancer patients with adenocarcinoma of the lung (see Chapter 5.2). Briefly, cancer patients from the Kentucky Lung Cancer Genomes (KLCG) cohort from the Appalachia region of Kentucky had the mutational frequencies of genes compared to patients from The Cancer Genome Atlas (59) cohort using the MutSigCV (27) protocol (see Chapter 5.2.1).

The MutSig dataset is a CSV file with one gene per row. Columns displaying the mutational frequency determined by MutSigCV for each cohort, KLCG and TCGA were also present, along with the p-value determined by comparing the mutational frequencies between the cohorts using a Fisher's exact test for each gene. We compiled foreground genes for the KLCG cohort by selecting genes that had a higher mutational frequency in the KLCG dataset than the TCGA dataset and that also had a p-value from the Fisher's exact test lower than 0.01.

Foreground genes were enriched against the universe, which was comprised of the whole set of genes in the dataset. This was performed using the enrichment methods described in Chapter 2.2. The only alteration needed for the enrichment workflows in this instance was changes to the configuration file: indicating the file paths to the data sets, and which p-value cutoffs to use, and a small, top-level script in the *snakefile*, dictating how foreground genes would be selected from the data set.

After enrichment was complete, additional rules were created for this analysis to accomplish the goals of grouping annotations by gene sets, creating tables to display these gene set-grouped annotations, and retrieving protein-protein interaction information from the STRING (55) database using STRING's REST application programming interface (API).

The first and second goals are accomplished by the rule, *group\_annotations\_by\_gene\_set*, which accepts the previously produced enrichment table as an input. Ignoring annotations with an odds ratio of “Inf” (meaning that only a single gene was annotated to the enriched term), a list of tuples is created, matching each enriched term to its set of associated genes. Meanwhile, all of the information from each row is saved in a dictionary, mapped to the enriched GO term ID. Next the list of tuples is sorted in order of the length of the associated gene set, largest first. Next, for each potential superset of genes in the list of tuples, the remaining gene sets are evaluated as to whether or not they are a subset of the set in the current iteration. If so, they are indicated as such and updated within the tuple list. The sorting of gene set length ensures that this process is as efficient as possible.

Once the gene supersets have been determined, a new output table is written by sorting the gene supersets by length again, and adding the appropriate rows from the saved dictionary, according to the annotations associated with the gene superset. These annotations are ordered in increasing value within each block of gene sets (see Table 5.2).

The third goal was accomplished by a rule called *retrieve\_interactions*. This rule is a simple script which makes URL requests to the STRING database via its REST interface to retrieve tabular information about known and predicted interactions for queried genes from the gene sets. For each gene superset identified by the previous step, a special URL string is formatted to request and download a TSV file showing the predicted and known interactions among the proteins queried, as well as additional nearest-neighbor proteins in the STRING database. To simplify this script’s placement in the overarching workflow, we enabled multiple tabular files to be output into a single document using the Pandas (60)

ExcelWriter Python module, where a new excel sheet would represent each interaction table, and an indexing sheet served to map the sheet numbers to the gene set query.

Images of the protein-protein interaction networks were produced by manually entering the query gene sets and selecting the species *Homo sapiens*. A single iteration of added nodes was performed by selecting “more” on the graphical user interface. We slightly modified of the nodes’ position within the network view to more clearly show edge coloration and to better proportion the image for print.

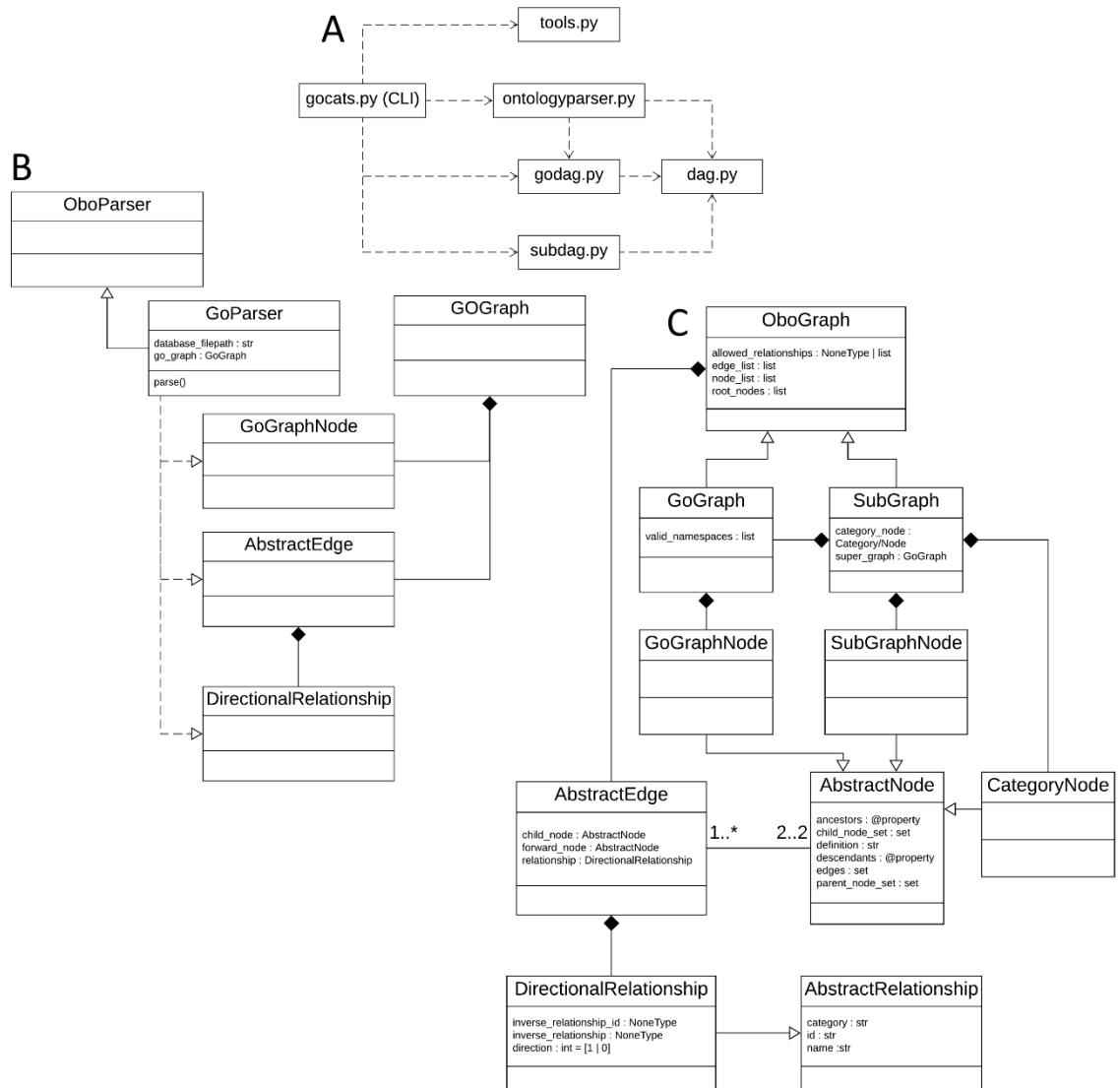


Figure 2.1 UML Diagrams Describing the GOcats Implementation

A) UML module dependency diagram. B) UML class diagram of GO database parsing classes. C) UML class diagram of the GO graph representation.

## CHAPTER 3. GOCATS: A TOOL FOR CATEGORIZING GENE ONTOLOGY INTO SUBGRAPHS OF USER-DEFINED CONCEPTS

### 3.1 Background

The Gene Ontology (GO) (3) is the most common biological controlled vocabulary (CV) used to represent information and knowledge distilled from most biological and biomedical research data generated today, from classic wet-bench experiments to high-throughput analytical platforms, especially omics technologies. The database structure of GO allows for computational retrieval of information by arranging biological terminology in a graph where nodes, representing terms, are connected by directional edges, representing relations that describe how terms are semantically related (see Chapter 1.2.2). These directional edges allow for semantic inferences to be made among the terms.

Differential abundance analyses for a range of omics-level technologies, especially transcriptomics technologies can yield large lists of differential genes, gene-products, or gene variants. In order to make sense of these large gene sets, researchers often rely on automated computational methods, such as annotation enrichment (see Chapter 1.3), to make sense of the data. However, such methods often fail to concisely summarize the biological concepts within the results, necessitating manual curation. This manual curation becomes more arduous as the size of data sets increase.

Previous efforts toward the organization of GO terms include GO Slims (32)—manually cut-down versions of GO to include only the more generalized terms, and Map2Slim (M2S) (13)—a tool which maps specific go terms to GO Slim terms using the graph structure of GO. The main limitations of M2S include are two-fold: categorization

is limited to the terms contained in the pre-compiled GO Slim, unless the user performs the time-consuming task of manually compiling lists of individual GO terms (essentially a custom GO Slim), and M2S requires the use of the GO-basic version of GO, which lacks many relations present in the GO-core version. By extension, we argue, the GO-core database is less informationally-rich. Additional information related to term categorization approaches is provided in Chapters 1.5 and 1.6.

In addition to organizing datasets and enrichment results, GO term categorization will likely serve a great benefit to ontology curators in maintaining and developing ontologies. As the field of information science grows, there is a greater need for the development and merging of ontologies to aid in the description of “big data” projects. Further information and examples are provided in Chapter 1.7

For the reasons indicated above, we have developed a tool called the GO Categorization Suite (GOcats), which serves to streamline the process of slicing the ontology into custom, biologically-meaningful subgraphs representing concepts derivable from GO. Unlike previously developed tools, GOcats uses a list of user-defined keywords and/or GO terms that describe a broad category-representative term from GO, along with the structure of GO and augmented relation properties to generate a subgraph of child terms and a mapping of these child terms to their respective category-defining term that is automatically identified based on the user’s keyword list, or to the GO term that is explicitly specified. Furthermore, these tools allow the user to choose between the strict axiomatic interpretation or a looser semantic scoping interpretation of part-whole (mereological) relation edges within GO. Specifically, we consider scoping relations to be comprised of

*is\_a*, *part\_of*, and *has\_part*, and mereological relations to be comprised of *part\_of* and *has\_part*.

## 3.2 Results

### 3.2.1 GOcats Compactly Organizes GO Subcellular Localization Terms into User-Specified Categories

As an initial proof-of-concept, we evaluated the automatic extraction and categorization of 25 subcellular locations, using GOcats’ “comprehensive” method of subgraph extension (see Chapter 2.1.1-2.1.3) and the GO-core graph (data-version: releases/2016-01-12). Starting with common biological subcellular concepts like “nucleus”, “cytoplasm”, and “mitochondrion”, we recursively used terms not being categorized to identify additional subcellular concepts and associated keywords represented within the GO Cellular Component sub-ontology. Due to the eventual application to the HPA datasets, three unusual categories, “bacterial”, “viral”, and “other organism”, were included to prevent categorization of terms that would complicate a eukaryotic interpretation of the other 22 subcellular locations. For these resulting 25 categories, 22 contained a designated GO term root-node that exactly matched the concept intended at the creation of the keyword list (Table 3.1).

These subgraphs account for approximately 89% of GO’s Cellular Component sub-ontology. While keyword querying of GO provided an initial seeding of the growing subgraph, Table 3.1 highlights the necessity of re-analyzing the GO graph, both to remove terms erroneously added by the keyword search and to add appropriate subgraph terms not captured by the keyword search. For example, the “cytoplasm” subgraph grew from its

initial seeding of 296 nodes to 1197 nodes after extension. Conversely, 136 nodes were seeded by keyword for the “bacterial” subgraph, but only 16 were rooted to the representative node.

To assess the relative size and structure of subgraphs within GO, we visualized the category subgraphs as a network using Cytoscape 3.0 (61). GOcats outputs a dictionary of individual GO term keys with a list of category-defining root-node values as part of its normal functionality.

Of note, 2102 of the 3877 terms in Cellular Component could be rooted to a single concept: “macromolecular complex.” Despite cytosol being defined as “the part of the cytoplasm that does not contain organelles, but which does contain other particulate matter, such as protein complexes”, less than half of the terms rooted to macromolecular complex also rooted to cytosol or cytoplasm. Surprisingly, approximately 25% of the terms rooted to macromolecular complex are rooted to this category alone (Figure 3.1). In this visualization, intracellular organelles tend to be clustered about cytoplasm, except for nucleus which the GO consortium does not consider as part of the cytoplasm. The visualization of the subgraph contents confirmed the uniqueness of the macromolecular complex category and showed the relative sizes of groups of GO terms shared between two or more categories. But the macromolecular complex category somewhat complicates the visualization of category organization within GO, due to this category’s size and interconnectedness within the ontology. To better reflect what might be a biologist’s expectation for a cell’s overall organization, we produced another visualization with the macromolecular complex category omitted (Figure 3.2). Despite the idiosyncrasies with the macromolecular complex subgraph, compartments that typically contain a large range

of protein complexes, such as the nucleus, plasma membrane, and cytoplasm appear to be appropriately populated. Furthermore, concepts such as endomembrane trafficking can be gleaned from the network connectedness of representative nodes, such as lysosome, Golgi apparatus, vesicle, secretory granule, and cytoplasm. Overall, the patterns of connectedness in this network make more sense biologically, within the constraints of GO's internal organization.

### 3.2.2 GOcats-derived Category Subgraphs Compare Well with Similar Subgraphs Derived by Other Methods

We compared GOcats' category subgraphs taken from the go-core database, data-version: releases/2016-01-12 to subgraphs of the manually-curated UniProt subcellular localization controlled vocabulary (CV) (24) (see Figure 3.2 and Chapter 3.4.1) and to subgraphs created by M2S (see Figure 3.3 and Chapter 3.4.2). Differences in the sets of GO terms contained within these subgraphs can be attributed to differences in the number of edges between nodes—as is the case between GOcats and M2S since M2S does not traverse across has\_part edges—and the number of overall nodes being evaluated—as is the case when comparing M2S and GOcats term sets to the UniProt CV terms sets since the UniProt CV contains considerably fewer GO terms. For the most part, GOcats category subgraphs are large supersets of UniProt CV subgraphs, as demonstrated by the high inclusion indices and low Jaccard indices in Table 3.2. In the comparison of GOcats and M2S subgraphs, the mappings for most categories are in very close agreement, as evidenced by both high inclusion and Jaccard indices in Table 3.3 and further highlighted in Figures 3.4A, 3.4B and Supplemental Figures 3.1 A-V (62). Overall, GOcats robustly

categorizes GO terms into category subgraphs with high similarity to existing GO-utilizing categorization methods while including information gleaned from has\_part edges.

However, in some categories, M2S and GOcats disagree as illustrated in Figure 3.4C and Supplemental Figure 3.1E. The most striking example of this is in the plasma membrane category, where M2S's subgraph contained over 300 terms that were not mapped by GOcats. We manually examined these discrepancies in the plasma membrane category and noted that many of the terms uniquely mapped by M2S did not appear to be properly rooted to "plasma membrane" (Supplemental Table 3.2). M2S mapped terms such as "nuclear envelope," "endomembrane system," "cell projection cytoplasm", and "synaptic vesicle, resting pool" to the plasma membrane category, while such questionable associations were not made using GOcats. Even though most terms included by M2S but excluded by GOcats exist beyond the scope of or are largely unrelated to the concept of "plasma membrane," a few terms in the set did seem appropriate, such as "intrinsic component of external side of cell outer membrane." However, of these examples, no logical semantic path could be traced between the term and "plasma membrane" in GO, indicating that these associations are not present in the ontology itself. These differences in mapping are due to our reevaluation of the has\_part edges with respect to scope. As shown in Table 3.3 the categories with the greatest agreement between the two methods were those with no instances of has\_part relations, which is the only relation in Cellular Component that is natively incongruent with respect to scope. However, there is no apparent correlation between the frequency of this relation and the extent of disagreement.

### 3.2.3 Custom-tailoring of GO Slim-like Categories with GOcats Allows for Robust Knowledgebase Gene Annotation Mining

The ability to query knowledgebases for genes and gene products related to a set of general concepts-of-interest is an important method for biologists and bioinformaticians alike. We hypothesized that grouping annotations into categories using GOcats and relevant keywords would more closely match the annotations categorized manually by the HPA consortium than either M2S or UniProt's CV. Using the set of GO terms annotated in the HPA's immunohistochemistry localization raw data as "concepts" (Table 3.4), we derived mappings to annotation categories generated from GOcats, M2S, and UniProt's CV based on UniProt- and Ensembl-sourced annotations from the European Molecular Biology Laboratories-European Bioinformatics Institute (EMBL-EBI) QuickGO knowledgebase resource (33) (See Chapter 3.4.5).

Next, we evaluated how these derived annotation categories matched raw HPA data GO annotations (See Chapter 3.4.5). GOcats slightly outperformed M2S and significantly outperformed UniProt's CV in the ability to query and extract genes and gene products from the knowledgebase that exactly matched the annotations provided by the HPA (Figure 3.5A). Similar relative results are seen for partially matched knowledgebase annotations. Genes in the "partial agreement," "partial agreement is superset," or "no agreement" groups may have annotations from other sources that place the gene in a location not tested by the HPA immunohistochemistry experiments or may be due to non-HPA annotations being at a higher semantic scoping than what the HPA provided. Also, novel localization

provided by the HPA could explain genes in the “partial agreement” and “no agreement” groups.

Furthermore, GOcats performed the categorization of HPA’s subcellular locations dataset in an average of 10.574 seconds after 50 test runs (standard deviation of 0.074 seconds), while M2S performed its mapping on the same data in an average of 14.837 seconds after 50 test runs (standard deviation of 0.300 seconds) (see Chapter 3.4.6 for hardware configuration details). These results are rather surprising since GOcats is implemented in Python (49), an interpreted language, versus M2S which is implemented in Java and compiled to Java byte code. However, the utilization of stored ancestor and descendent node sets facilitated the implementation of efficient subgraph-centric algorithms within GOcats. Based on these results, GOcats should offer appreciable computational improvement on significantly larger datasets.

One key feature of GOcats is the ability to easily customize category subgraphs of interest. To improve agreement and rectify potential differences in term granularity, we used GOcats to organize HPA’s raw data annotation along with the knowledgebase data into slightly more generic categories (Table 3.5). In doing so, GOcats can query over twice as many knowledgebase-derived gene annotations with complete agreement with the more-generic HPA annotations, while also increasing the number of genes in the categories of “partial” and “partial agreement is superset” agreement types and decreasing the number of genes in the “no agreement” category (Figure 3.5B).

We then compared the methods’ mapping of knowledgebase gene annotations derived from HPA to the HPA experimental dataset to demonstrate how researchers could use the GOcats suite to evaluate how well their own experimental data is represented in

public knowledgebases. Because the set of gene annotations used in the HPA experimental dataset and in the HPA-derived knowledgebase annotations are identical, no term mapping occurred during the agreement evaluation and so the assignment agreement was identical between GOcats and M2S. As expected, the complete agreement category was high, although there was a surprising number of partial agreement and even some genes that had no annotations in agreement (Figure 3.5A). We next broke down which locations were involved in each agreement type and noted that the “nucleus,” “nucleolus,” and “nucleoplasm” had the highest disagreement relative to their sizes, but these disagreements were present across nearly all categories (Table 3.5).

Both M2S and GOcats avoid superset category term mapping; neither map a category-representative GO term to another category-representative GO term if one supersedes another (although GOcats has the option to enable this functionality). Therefore, discrepancies in annotation should not arise by term mapping methods. Nevertheless, we hypothesized that some granularity-level discrepancies exist between the HPA experimental raw data and the HPA-assigned gene annotations in the knowledgebase. We performed the same custom category generic mapping as we did for the previous test and discovered that some disagreements were indeed accounted for by granularity-level discrepancies, as seen in the decrease in “partial” and “no agreement” categories and increase in “complete” agreement category following generic mapping (Figure 3.6, blue bars). For example, 26S proteasome non-ATPase regulatory subunit 3 (PSMD3) was annotated to the nucleus (GO:0005634) and cytoplasm (GO:0005737) in the experimental data but was annotated to the nucleoplasm (GO:0005654) and cytoplasm in the knowledgebase. By matching the common ancestor mapping term “nucleus”, GOcats can

group the two annotations in the same category. In total, 132 terms were a result of semantic scoping discrepancies. Worth noting is the fact that categories could be grouped to common categories to further improve agreement, for example “nucleolus” within “nucleus.”

Interestingly, among the remaining disagreeing assignments were some with fundamentally different annotations. Many of these are cases in which either the experimental data, or knowledgebase data have one or more additional locations distinct from the other. For example, NADH dehydrogenase [ubiquinone] 1 beta subcomplex subunit 6 (NDUB6) was localized only to the mitochondria (GO:0005739) in the experimental data yet has annotations to the mitochondria and the nucleoplasm (GO:0005654) in the knowledgebase. Why such discrepancies exist between experimental data and the knowledgebase is not clear.

We were also surprised by the high number of genes with “supportive” annotations in the HPA raw data that were not found in the EMBL-EBI knowledgebase when filtered to those annotated by HPA. As Figure 3.6 shows, roughly one-third of the annotations from the raw data were missing altogether from the knowledgebase; the gene was not present in the knowledgebase whatsoever. This was surprising because “supportive” was the highest confidence score for subcellular localization annotation.

### 3.3 Discussion and Conclusions

Discrepancies in the semantic granularity of gene annotations in knowledgebases represent a significant hurdle to overcome for researchers interested in mining genes based on a set of annotations used in experimental data. To demonstrate the potential GOcats has in resolving these discrepancies, we categorized annotations from HPA-sourced gene annotations using GOcats, M2S, and the UniProt subcellular localization CV. The HPA

source was chosen because primary data from high-throughput immunofluorescence-based gene product localization experiments exist in publicly-accessible repositories and have been inspected by experts and given a confidence score (31). As we show, utilizing only the set of specific annotations used in the HPA's experimental data, M2S's mapping matches only 366 identical sets of gene annotations from the knowledgebase with GOcats matching slightly more (Figure 3.5A). GOcats alleviates this problem by allowing researchers to define categories at a custom level of granularity so that categories may be specific enough to retain biological significance, but generic enough to encapsulate a larger set of knowledgebase-derived annotations. When we reevaluated the agreement between the raw data and knowledgebase annotations using custom GOcats categories for "cytoskeleton" and "nucleus", the number of identical gene annotations increased to 776 (Figure 3.5B).

Because GOcats relies on user-input keywords to define categories, we understand that there is a risk of adding user bias when applying this method to organizing results of various analyses. While we have taken care to avoid bias in the comparisons made in this report, for example citing the exact category defining GO term for each category compared between methods (Figure 3.4, Table 3.2, Table 3.3) and reporting the exact common-sense categorizations applied when grouping location categories from HPA (Table 3.5), we strongly caution users to exercise similar care in their use as well. For instance, when categorizing results from annotation enrichment analyses it may be tempting to filter results to those categories defined by the user, which might conveniently eliminate unexpected (unwanted) highly-enriched terms. We do not condone the use of GOcats in this way. But because GOcats will always produce the same subgraph categorizations for the same set of

keywords used with the same version of GO, we argue that our categorization is more reproducible and less prone to bias than manually grouping GO terms into categories or otherwise manually identifying major concepts represented from omics-level analyses.

As GO continues to grow, automated methods to evaluate the structural organization of data will become necessary for curation and quality control. Because GOcats allows versatile interpretation of the GO directed acyclic graph (DAG) structure, it has many potential curation and quality control uses, especially for evaluating the high-level ontological organization of GO terms. For example, GOcats can facilitate the integrity checking of annotations that are added to public repositories by streamlining the process of extracting categories of annotations from knowledgebases and comparing them to the original annotations in the raw data. Interestingly, about one-third of the genes annotated with high-confidence in the HPA raw data were missing altogether from the EMBL-EBI knowledgebase when filtered to the HPA-sourced annotations. While this surprised us, the reason appears to be due to HPA's use of two separate criteria for "supportive" annotation reliability scores and for knowledge-based annotations. For "supportive" reliability, one of several conditions must be met: i) two independent antibodies yielding similar or partly similar staining patterns, ii) two independent antibodies yielding dissimilar staining patterns, both supported by experimental gene/protein characterization data, iii) one antibody yielding a staining pattern supported by experimental gene/protein characterization data, iv) one antibody yielding a staining pattern with no available experimental gene/protein characterization data, but supported by other assay within the HPA, and v) one or more independent antibodies yielding staining patterns not consistent with experimental gene/protein characterization data, but supported by siRNA assay (31)

Meanwhile knowledge-based annotations are dependent on the number of cell lines annotated; specifically, the documentation states, “Knowledge-based annotation of subcellular location aims to provide an interpretation of the subcellular localization of a specific protein in at least three human cell lines. The conflation of immunofluorescence data from two or more antibody sources directed towards the same protein and a review of available protein/gene characterization data, allows for a knowledge-based interpretation of the subcellular location” (31). Unfortunately, we were unable to explore these differences further, since the experimental data-based subcellular localization annotations appeared aggregated across multiple cell lines, without specifying which cell lines were positive for each location. Meanwhile, tissue- and cell-line specific data, which contained expression level information, did not also contain subcellular localizations. Therefore, we would suggest that HPA and other major experimental data repositories always provide a specific annotation reliability category in their distilled experimental datasets that matches the criteria used for deposition of derived annotations in the knowledgebases. Such information will be invaluable for performing knowledgebase-level evaluation of large curated sets of annotations. One step better would involve providing a complete experimental and support data audit trail for each derived annotation curated for a knowledgebase, but this may be prohibitively difficult and time-consuming to do.

Looking towards the future, the work demonstrated here is a critical first step towards a goal of automatically enumerating all representable concepts within GO. Such an enumeration would provide scientists with the usable set of GO-representable concept subgraphs for a large variety of analyses unbiased by human selection. GOcats can derive subgraphs representing a specific concept by utilizing keywords and key terms, which

would be a major component for an overall method to enumerate all representable concepts. We expect two other major components will be required, first is a way to derive possible keywords and key terms and the last is a way to evaluate the quality of the concept subgraphs that are generated. We expect the latter evaluation to involve the development of various graph-based metrics for this purpose. Details for these future developments are provided in chapter 6.

In this study, we: i) demonstrated an improvement in retrievable ontological information content by the reevaluation of GO's `has_part` relation ii) applied our new method GOcats toward the categorization and utilization of the GO Cellular Component sub-ontology, and iii) evaluated the ability of GOcats and other mapping tools to relate HPA experimental to HPA knowledgebase GO Cellular Component annotation sources. GOcats outperforms the UniProt CV with respect to accurately deriving gene-product subcellular location from the UniProt and Ensembl database with the HPA raw dataset of gene localization annotations treated as the gold standard. Moreover, GOcats comparison to M2S demonstrates similar mapping performance between the two methods, but with GOcats providing important improvements in mapping, computational speed, ease of use, and flexibility of use.

In conclusion, GOcats enables the user to create custom, GO slim-like filters to map fine-grained gene annotations from GAFs to general subcellular compartments without needing to hand-select a set GO terms for categorization. Moreover, users can use GOcats to quickly customize the level of semantic specificity for annotation categories. Furthermore, GOcats was designed for scientists who are less familiar with GO. GOcats enables a safe and more comprehensive semantic scoping utilization of go-core, preventing

mistakes that can easily arise from using go-core instead of go-basic. Together, these improvements will impact a variety of GO knowledgebase data mining use-cases as well as knowledgebase curation and quality control. Looking towards the future, GOcats provides a critical categorization method for a future automatic enumeration of all representable concepts within GO.

### 3.4 Methods

#### 3.4.1 Creating Category Mappings from UniProt's Subcellular Location Controlled Vocabulary

We created mappings from fine-grained to general locations in UniProt's subcellular location CV (14) for comparison to GOcats. To accomplish this, we parsed and recreated the graph structure of UniProt's subcellular locations CV file (24) in a manner similar to the parsing of GO. Briefly, the flat-file representation of the CV file is parsed line-by-line and each term is stored in a dictionary along with information about its graph neighbors as well as its cross-referenced GO identifier. We assumed that terms without parent nodes in this graph are category-defining root-nodes and created a dictionary where a root-node key links to a list of all recursive children of that node in the graph. Only those terms with cross-referenced GO identifiers were included in the final mapping. The category subgraphs created from UniProt were compared to those with corresponding category root-nodes made by GOcats. An inclusion index,  $I$ , was calculated by considering the two subgraphs' members as sets and applying the following equation:

$$I = \frac{|S_n \cap S_g|}{|S_n|} \quad (1)$$

where  $S_n$  and  $S_g$  are the set of members within the non-GOcats-derived category and GOcats-derived category, respectively. It is worth noting here that the size of the UniProt set was always smaller than the GOcats set. This is due to the inherent size differences between UniProt's CV and the Cellular Component sub-ontology.

### 3.4.2 Creating Category Mappings from Map2Slim

The Java implementation of OWLTools' M2S does not include the ability to output a mapping file between fine-grained GO terms and their GO slim mapping target from the GAF that is mapped. To compare subgraph contents of GOcats categories to a comparable M2S "category," we created a special custom GAF where the gene ID column and GO term annotation column of each line were each replaced by a different GO term for each GO term in Cellular Component, data-version: releases/2016-01-12. We then allowed M2S to map this GAF with a provided GO slim. The resulting mapped GAF was parsed to create a standalone mapping between the terms from the GO slim and a set of the terms in their subgraphs.

### 3.4.3 Mapping Gene Annotations to User-defined Categories

To allow users to easily map gene annotations from fine-grained annotations to specified categories, we added functionality for accepting GAFs as input, mapping annotations within the GAF and outputting a mapped GAF into a user-specified results directory. The input-output scheme used by GOcats and M2S are similar, with the exception that GOcats accepts the mapping dictionary created from category keywords, as described previously, instead of a GO slim. GAFs are parsed as a tab-separated-value file.

When a row contains a GO annotation in the mapping dictionary, the row is rewritten to replace the original fine-grained GO term with the corresponding category-defining GO term. If the gene annotation is not in the mapping dictionary, the row is not copied to the mapped GAF, and is added to a separate file containing a list of unmapped genes for review. The mapped GAF and list of unmapped genes are then saved to the user-specified results directory.

#### 3.4.4 Visualizing and Characterizing Intersections of Category Subgraphs

To compare the contents of category subgraphs made by GOcats, UniProt CV, and M2S, we took the set of subgraph terms for each category in each method, converted them into a Pandas DataFrame (60) representation, and plotted the intersections using the UpSetR R package (62). Inclusion indices were also computed for M2S categories using Equation 1. Jaccard indices were computed for every subgraph pair to evaluate the similarity between subgraphs of the same concept, created by different methods.

#### 3.4.5 Assigning Generalized Subcellular Locations to Genes from the Knowledgebase and Comparing Assignments to Experimentally-Determined Locations

We first mapped two GAFs downloaded from the EMBL-EBI QuickGO resource (33) using GOcats, the UniProt CV, and M2S. We filtered the gene annotations by dataset source and evidence type, resulting in separate GAFs containing annotations from the following sources: UniProt-Ensembl, and HPA. Both GAFs had the evidence type, Inferred from Electronic Annotation (IEA), filtered out because it is generally considered to be the least reliable evidence type for gene annotation and in the interest of minimizing memory

usage. We used this data to assess the performance of the mapping methods in their ability to assign genes to subcellular locations based on annotations from knowledgebases by comparing these assignments to those made experimentally in HPA's localization dataset (Figure 3.5A). Comparison results for each gene were aggregated into 4 types: i) "complete agreement" for genes where all subcellular locations derived from the knowledgebase and the HPA dataset matched, ii) "partial agreement" for genes with at least one matching subcellular location, iii) "partial superset" for genes where knowledgebase subcellular locations are a superset of the HPA dataset, iv) "no agreement" for genes with no subcellular locations in common, and v) "no annotations" for genes in the experimental dataset that were not found in the knowledgebase.

Only gene product localizations from the HPA dataset with a "supportive" confidence score were used for this analysis (n=4795). We created a GO slim by looking up the corresponding GO term for each location in this dataset with the aid of QuickGO term basket and filtering tools. The resulting GO slim served as input for the creation of mapped GAFs using M2S. To create mapped GAFs using GOcats, we entered keywords related to each location in the HPA dataset (Table 3.4). We matched the identifier in the "gene name" column of the experimental data with the identifier in the "database object symbol" column in the GAF to compare gene annotations. Our assessment of comparing the HPA raw data to mapped gene annotations from the knowledgebase represents the ability to accurately query and mine genes and their annotations from the knowledgebase into categories of biological significance. Our assessment of comparing the methods' mapping output to the HPA raw dataset represents the ability of these methods to evaluate the representation of HPA's latest experimental data as it exists in public repositories.

### 3.4.6 Running Time Tests between GOcats and Map2Slim Categorizations

For comparing the runtimes of GOcats and M2S for categorizing HPA's subcellular location dataset, each method was run separately on the same machine with the following configuration: Intel ® Core ™ i7-4930K CPU with 6 hyperthreaded cores clocked at 3.40GHz and 64 GB of RAM clocked at 1866 MHz. We used the Linux "time" command with no additional options and reported the real time from its output. The datasets and scripts used for this evaluation have been uploaded to a FigShare repository (63). We used the dataset contained in our ref 63: KBDData/11-02-2016/hpa-no\_IEA.goa for these comparisons. For M2S we executed a custom script that can be found within ref 63: runscripts:

```
$ sh owlmultitest.sh
```

which ran the following command, found in the same subdirectory, 50 times:

```
$ time sh owltoolsspeedtest.sh
```

For GOcats, we executed a custom script that can be found within ref 63, runscripts:

```
$ sh gcmultitest.sh
```

which ran the following command, found in the same subdirectory, 50 times:

```
$ time sh GOcatsspeedtest.sh
```

Both tests were executed using the same version of the go-core, which is data version: releases/2016-01-12 (63).

Table 3.1 Summary of 25 Example Subcellular Locations Extracted by GOcats

Subgraph name	User-input keywords	Predicted representative term (ID)	Nodes seeded from keyword search	Nodes added during graph extension	Seeded nodes not in subgraph	Total nodes
Aggresome	aggresome, aggresomal, aggresomes	aggresome (GO:0016235)	1	0	0	1
Bacterial	bacterial, bacteria, bacterial-type	bacterial-type flagellum (GO:0009288)	136	1	121	16
Cell Junction	junction	Cell junction (GO:0030054)	68	16	34	50
Chromosome	chromosome, chromosomal, chromosomes	chromosome (GO:0005694)	120	122	31	211
Cytoplasm	cytoplasm, cytoplasmic	Cytoplasm (GO:0005737)	296	1061	160	1197
Cytoplasmic Granule	granule, granules	secretory granule (GO:0030141)	81	16	50	47
Cytoskeleton	cytoskeleton, cytoskeletal	cytoskeleton (GO:0005856)	78	194	47	225
Cytosol	cytosol, cytosolic	cytosol (GO:0005829)	56	51	28	79
Endoplasmic Reticulum	endoplasmic, sarcoplasmic, reticulum	endoplasmic reticulum (GO:0005783)	113	39	51	101

Endosome	endosome, endosomes, endosomal	endosome (GO:0005768)	67	15	24	58
Extracellular	extracellular, secreted	extracellular region (GO:0005576)	142	123	85	180
Golgi Apparatus	golgi	golgi apparatus (GO:0005794)	67	12	25	54
Lysosome	lysosome, lysosomal, lysosomes	lysosome (GO:0005764)	42	7	16	33
Macromolecular Complex	protein, macromolecular	macromolecular complex (GO:0032991)	1317	969	184	2102
Microbody	microbody, microbodies	microbody (GO:0042579)	4	20	0	24
Mitochondrion	mitochondria, mitochondrial, mitochondrion	mitochondrion (GO:0005739)	134	2	44	92
Neuron Part	neuron, neuronal, neurons, synapse	neuron part (GO:0097458)	90	94	35	149
Nucleolus	nucleolus, nucleolar	nucleolus (GO:0005730)	25	11	12	24
Nucleus	nucleus, nuclei, nuclear	nucleus (GO:0005634)	288	340	118	510
Other Organism	other, host, organism	other organism (GO:0044215)	369	12	259	122

Plasma Membrane	plasma	plasma membrane (GO:0005886)	308	302	164	446
Plastid	plastid, chloroplast	plastid (GO:0009536)	95	48	8	135
Thylakoid	thylakoid, thylakoids	thylakoid (GO:0009579)	52	22	11	63
Vesicle	vesicle, vesicles	vesicle (GO:0031982)	198	90	85	203
Viral	virion, virus, viral	viral occlusion body (GO:0039679)	93	1	26	68
	Expected representative					
	Unexpected representative					

Table 3.2 Agreement Summary between Corresponding GOcats and UniProt CV Subgraphs

Location Category	Term ID	Inclusion Index	Jaccard Index	GOcats subgraph size	UniProt CV subgraph size
Bacterial-type Flagellum	GO:0009288	1	0.0625	16	1
Cell Junction	GO:0030054	0.47619	0.163934	50	21
Chromosome	GO:0005694	1	0.0189573	211	4
Cytoplasm	GO:0005737	0.809524	0.0141549	1197	21
Endoplasmic Reticulum	GO:0005783	0.818182	0.0873786	101	11
Endosome	GO:0005783	1	0.241379	58	14
Extracellular Region	GO:0005576	0.5625	0.0481283	180	16
Golgi Apparatus	GO:0005794	0.8	0.142857	54	10
Lysosome	GO:0005764	1	0.0909091	33	3
Mitochondrion	GO:0005739	1	0.0978261	92	9
Nucleus	GO:0005634	1	0.0294118	510	15
Plastid	GO:0009536	0.846154	0.307692	135	52

Table 3.3 Agreement Summary between Corresponding GOcats and Map2Slim Subgraphs

Location Category	Term ID	Inclusion Index <sup>‡</sup>	Jaccard Index	GOcats subgraph size	Map2Slim subgraph size	"Has_part" relationships
Aggresome	GO:0016235	1	1	1	1	0
Bacterial-type Flagellum	GO:0009288	1	1	16	16	8
Cell Junction	GO:0030054	0.980392	0.980392	50	51	4
Chromosome	GO:0005694	0.984375	0.883178	211	192	40
Cytoplasm	GO:0005737	0.927273	0.452055	1197	605	38
Cytoskeleton	GO:0005856	0.812274	0.812274	225	277	10
Cytosol	GO:0005829	0.963415	0.963415	79	82	8
Endoplasmic Reticulum	GO:0005783	1	0.990099	101	100	4
Endosome	GO:0005768	1	1	58	58	0
Extracellular Region	GO:0005576	1	0.927778	180	167	2
Golgi Apparatus	GO:0005794	1	1	54	54	0
Lysosome	GO:0005764	1	1	33	33	0
Macromolecular Complex	GO:0032991	0.947274	0.947274	2102	2219	232
Microbody	GO:0042579	1	1	2	24	0
Mitochondrion	GO:0005739	0.978723	0.978723	92	94	8
Neuron Part	GO:0097458	1	0.993289	149	148	22
Nucleolus	GO:0005730	0.857143	0.857143	24	28	0
Nucleus	GO:0005634	0.991684	0.928016	510	481	168
Other Organism	GO:0044215	1	1	122	122	8
Plasma Membrane	GO:0005886	0.563081	0.547097	446	753	20
Plastid	GO:0009536	0.992647	0.992647	135	136	0

Secretory Granule	GO:0030141	1	1	47	47	0
Thylakoid	GO:0009579	1	1	63	63	0
Vesicle	GO:0031982	0.981132	0.757282	203	159	12
Viral Occlusion Body	GO:0039679	1	0.0147059	68	1	4

‡ Inclusion index quantifies the extent to which the smaller subgraph is included in the larger subgraph

Table 3.4 Summary of 20 Subcellular Locations Used in the HPA Raw Experimental Data Extracted by GOcats

Subgraph name	User-input keywords	Predicted representative term (ID)	Nodes seeded from keyword search	Nodes added during graph extension	Seeded nodes not in subgraph	Total nodes
Actin cytoskeleton	actin cytoskeleton	actin cytoskeleton (GO:0015629)	117	22	77	62
Aggresome	aggresome, aggresomal, aggresomes	aggresome (GO:0016235)	1	0	0	1
Cell Junction	junction	cell junction (GO:0030054)	68	16	34	50
Centrosome	centrosome	centrosome (GO:0005813)	10	2	5	7
Cytoplasm	cytoplasm, cytoplasmic	cytoplasm (GO:0005737)	296	1061	160	1197
Endoplasmic Reticulum	endoplasmic, sarcoplasmic, reticulum	endoplasmic reticulum (GO:0005783)	113	39	51	101
Focal adhesion	focal adhesion	focal adhesion (GO:0005925)	29	0	28	1
Golgi Apparatus	golgi	golgi apparatus (GO:0005794)	67	12	25	54
Intercellular bridge	intercellular bridge	intercellular bridge (GO:0045171)	24	2	19	7
Intermediate filament cytoskeleton	intermediate filament cytoskeleton	intermediate filament cytoskeleton (GO:0045111)	126	0	118	8

Intracellular membrane-bounded organelle (vesicle <sup>‡</sup> )	intracellular membrane-bounded organelle	Intracellular membrane-bounded organelle (GO:0043231)	229	1116	118	1227
Microtubule cytoskeleton	microtubule cytoskeleton	microtubule cytoskeleton (GO:0015630)	112	55	68	109
Microtubule end	microtubule end	microtubule end (GO:1990752)	138	0	133	5
Microtubule organizing center	microtubule organizing center	microtubule organizing center (GO:0005815)	110	34	95	49
Mitochondrion	mitochondria, mitochondrial, mitochondrion	mitochondrion (GO:0005739)	134	2	44	92
Nuclear membrane	nuclear membrane	nuclear membrane (GO:0031965)	1151	0	1139	12
Nucleolus	nucleolus, nucleolar	nucleolus (GO:0005730)	25	11	12	24
Nucleoplasm	nucleoplasm	nucleoplasm (GO:0005654)	10	125	4	131
Nucleus	nucleus, nuclei, nuclear	nucleus (GO:0005634)	288	340	118	510
Plasma Membrane	plasma	plasma membrane (GO:0005886)	308	302	164	446

<sup>‡</sup> HPA conservatively annotates "vesicles" as intracellular membrane-bounded organelle

	Expected representative
	Unexpected representative

Table 3.5 Generic Location Categories Used to Resolve Potential Scoping Inconsistencies in HPA Raw Data

<b>HPA annotation category</b>	<b>GOcats-customized general HPA category</b>
Actin cytoskeleton	Cytoskeleton
Centrosome	
Intermediate filament cytoskeleton	
Microtubule cytoskeleton	
Microtubule end	
Microtubule organizing center	
Aggresome	Aggresome
Cell junction	Cell junction
Cytoplasm	Cytoplasm
Endoplasmic reticulum	Endoplasmic reticulum
Focal adhesion	Focal adhesion
Golgi apparatus	Golgi apparatus
Intercellular bridge	intercellular bridge
intracellular membrane-bounded organelle	intracellular membrane-bounded organelle
Mitochondrion	Mitochondrion
Nucleus	Nucleus
Nucleoplasm	
Nuclear membrane	
Nucleolus	Nucleolus
Plasma membrane	Plasma membrane

Table 3.6 Summary of Gene Location Category Agreement between Manually-curated HPA Raw Data and GOCats/Map2Slim Categorized HPA-derived Annotations

	<b>Agreement</b>				
<b>Location</b>	<b>Complete</b>	<b>Partial</b>	<b>Superset<sup>‡</sup></b>	<b>None</b>	<b>Not in Knowledgebase</b>
Actin cytoskeleton	51	0	7	0	37
Aggresome	2	0	0	3	4
Cell Junction	36	0	17	0	51
Centrosome	58	3	17	0	49
Cytoplasm	1037	55	162	5	643
Endoplasmic Reticulum	66	1	7	0	39
Focal adhesion	27	5	9	0	17
Golgi Apparatus	159	5	43	0	137
Intercellular bridge	14	0	4	0	19
Intermediate filament cytoskeleton	18	1	4	0	23
Intracellular membrane-bounded organelle	283	6	50	1	212
Microtubule cytoskeleton	35	2	9	0	27
Microtubule end	2	0	0	0	0
Microtubule organizing center	32	0	5	0	14
Mitochondrion	263	4	55	0	154
Nuclear membrane	47	6	17	0	39
Nucleolus	266	10	69	6	163
Nucleoplasm	989	26	230	23	534
Nucleus	437	14	217	23	373
Plasma Membrane	265	12	55	0	225

<sup>‡</sup>Knowledgebase genes mapped to a set of categories that is a superset of those manually assigned by the HPA in raw data

\* Numbers reflect how many times a location was involved in a particular agreement type; sums of all locations for an agreement category do not indicate the total number of genes for an agreement type.

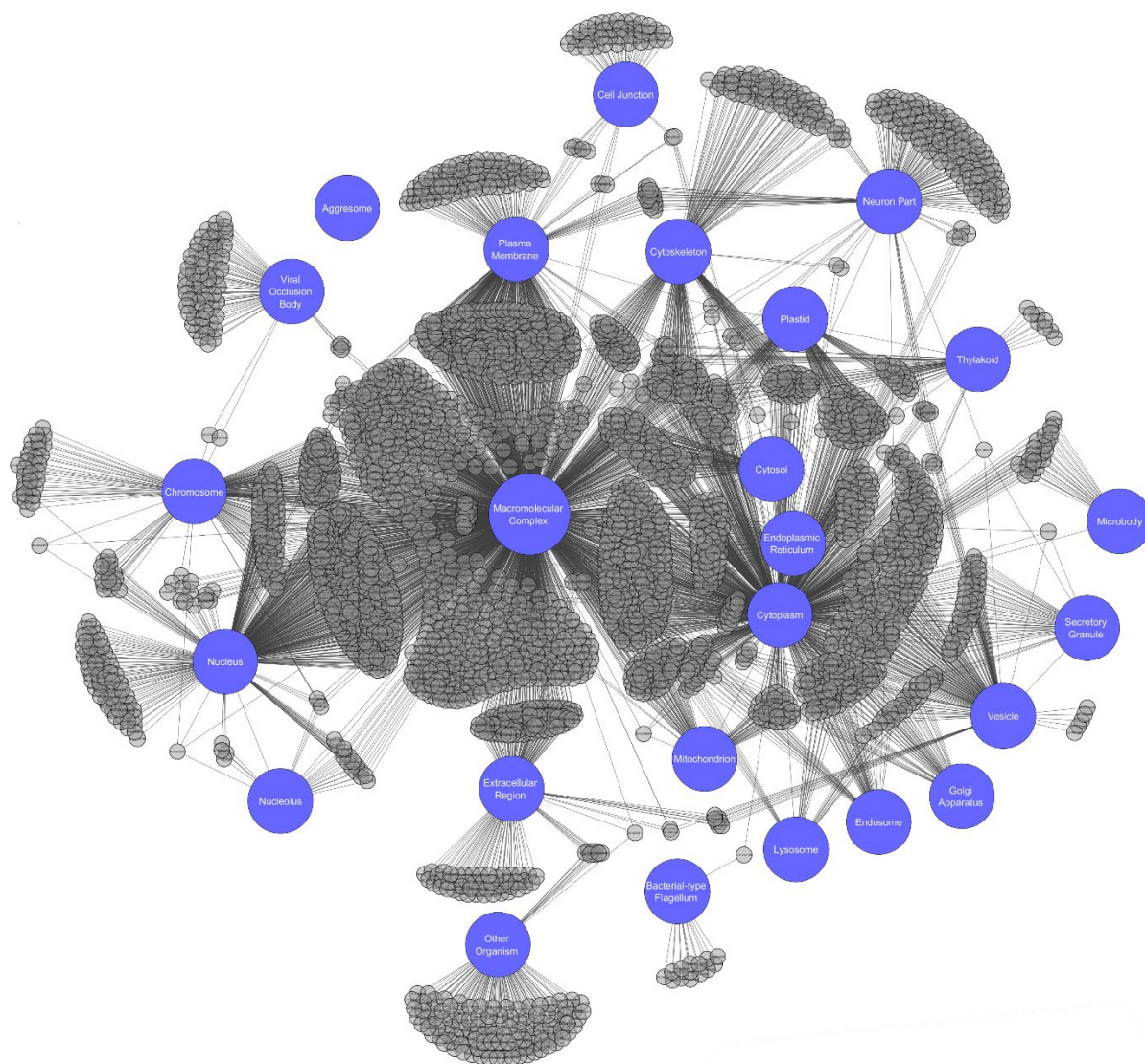


Figure 3.1 Network of 25 Categories Whose Subgraphs Account for 89% of the GO Cellular Component Sub-ontology

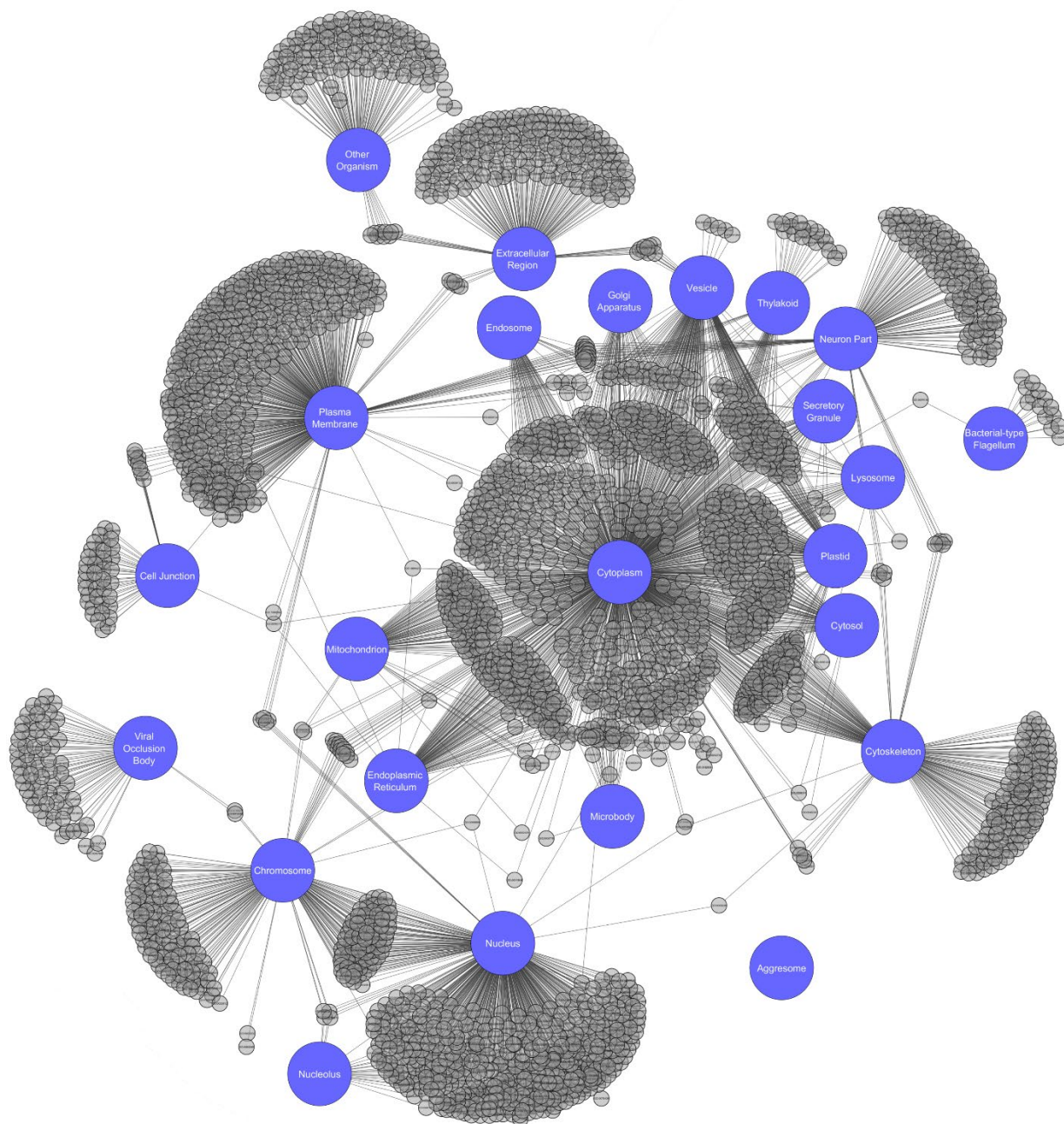


Figure 3.2 Network of All Categories from Figure 3.1 Except for Macromolecular Complex

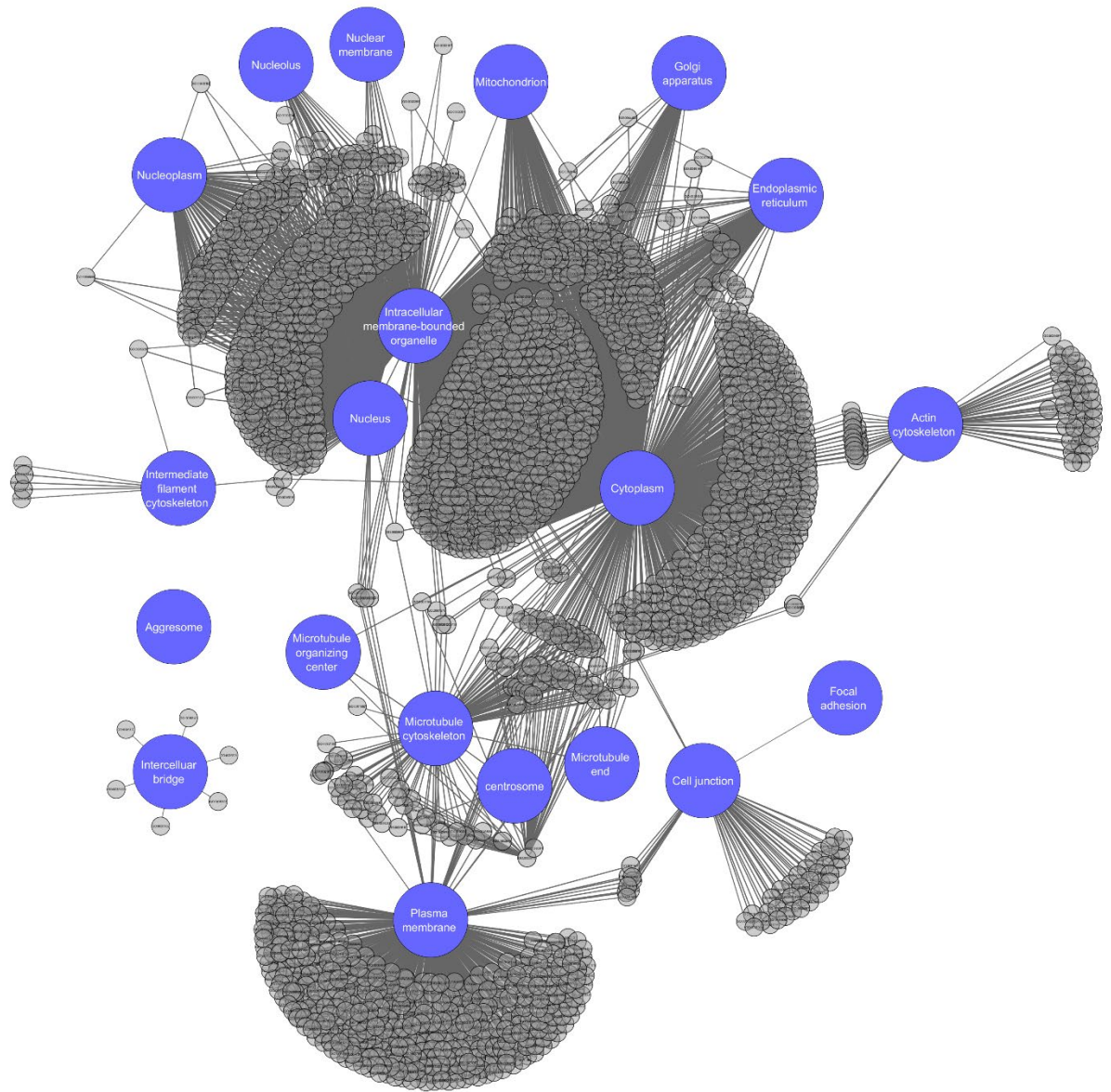


Figure 3.3 Network of 20 Categories Used in the Human Protein Atlas Subcellular Localization Immunohistochemistry Raw Data

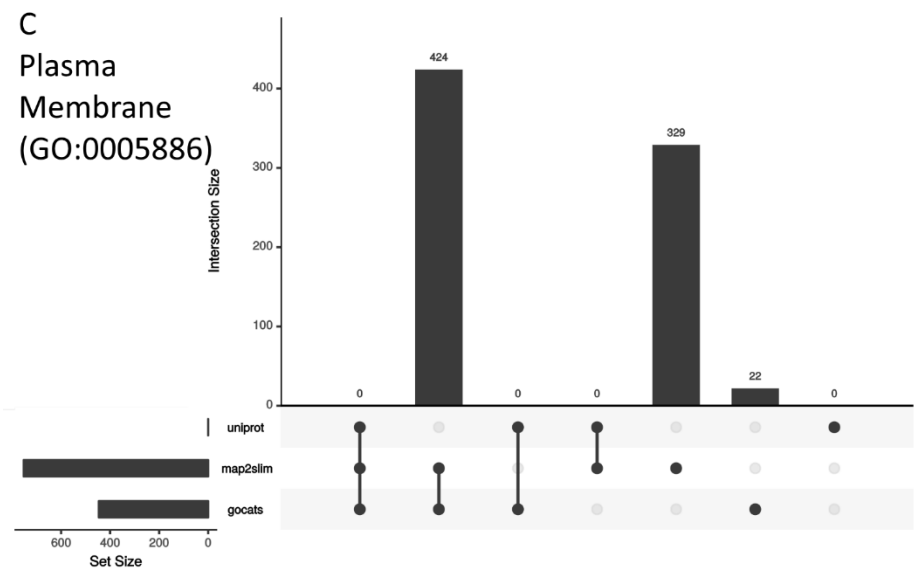
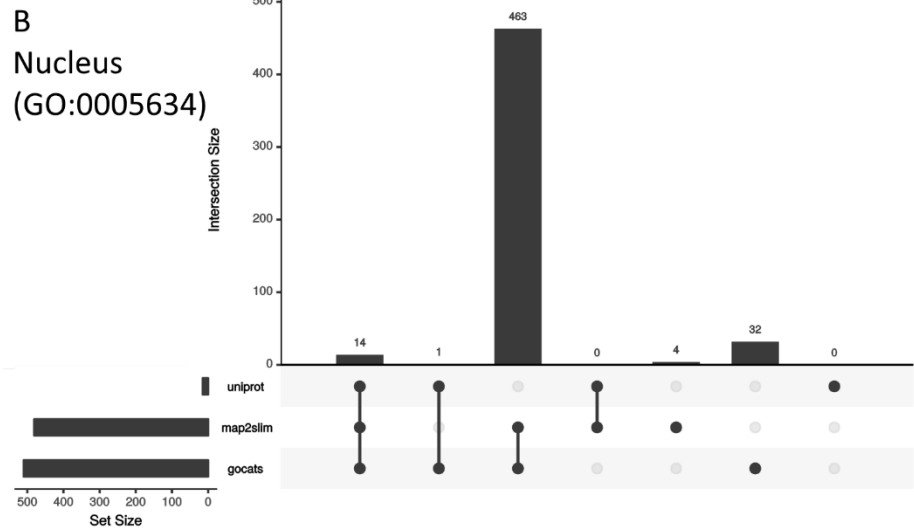
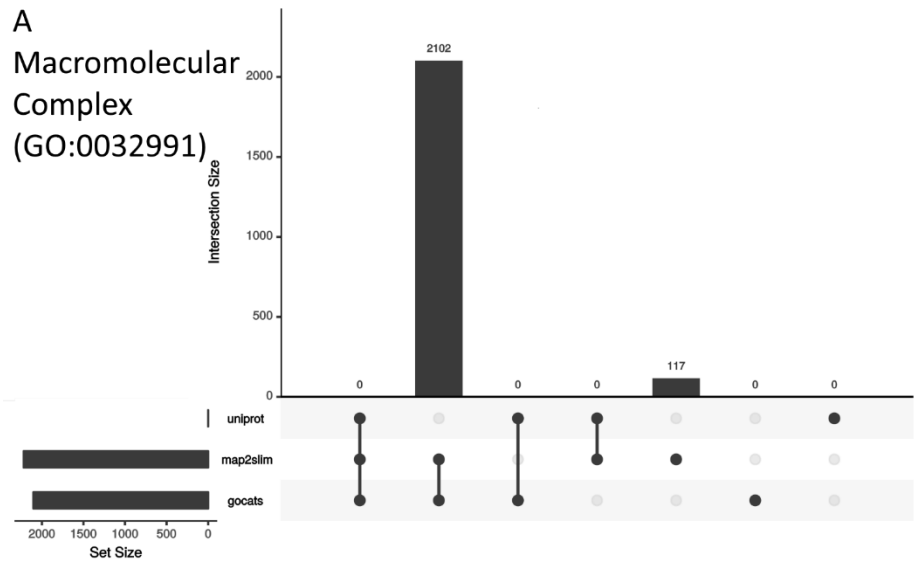


Figure 3.4 (continued) Visualizing the degree of overlap between the category subgraphs created by GOcats, Map2Slim, and the UniProt CV.

Plots were created using the R package: UpSetR (64), as a visual alternative to a Venn diagram. The amount of overlap between category-specific subgraphs are indicated by the vertical bar graph with the connect dots identifying which specific mapping method (UniProt, GOcats, and Map2Slim) is included in the overlap.

A) Macromolecular Complex; B) Nucleus; C) Plasma Membrane. Plots for all categories can be found in Supplemental Figures 3.1A-Y.

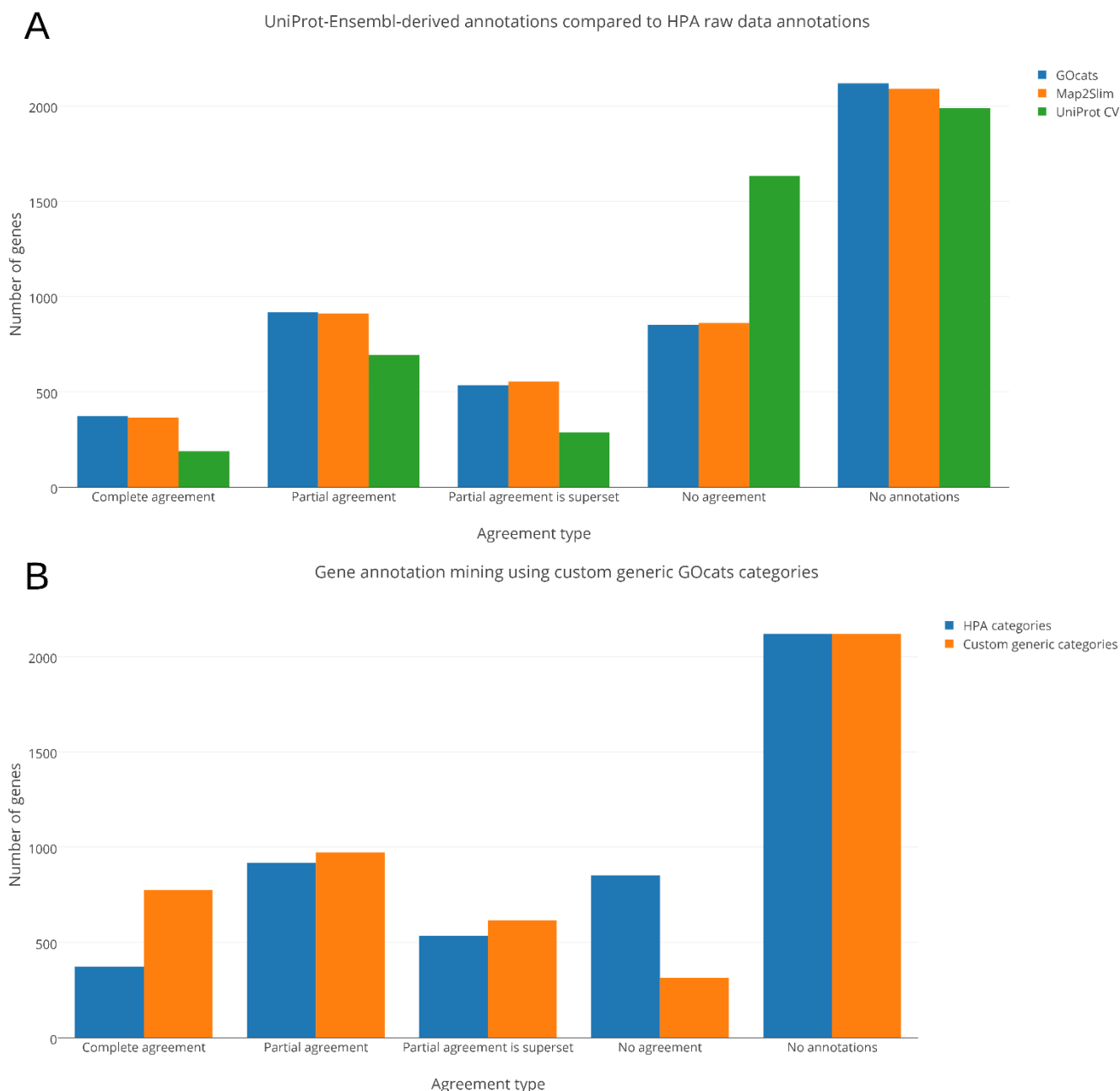


Figure 3.5. Comparison of UniProt-Ensembl knowledgebase annotation data mining extraction performance by GOcats, Map2Slim, and UniProt CV

“Complete agreement” refers to genes where all subcellular locations derived from the knowledgebase and the HPA dataset matched. “partial agreement” refers to genes with at least one matching subcellular location. “partial agreement is superset” refers to genes where knowledgebase subcellular locations are a superset of the HPA dataset (these are mutually exclusive to the “partial agreement” category).

Figure 3.5(continued) "no agreement" refers to genes with no subcellular locations in common. "no annotations" refers to genes in the experimental dataset that were not found in the knowledgebase. The more-generic categories used in panel B can be found in Table 3.5.

A) Number of genes of the given agreement type when comparing mapped gene product annotations assigned by UniProt and Ensembl in the EMBL-EBI knowledgebase to those taken from The Human Protein Atlas' raw data. Knowledgebase annotations were mapped by GOcats, Map2Slim, and the UniProt CV to the set of GO annotations used by the HPA in their experimental data. B) Shift in agreement following GOcats' mapping of the same knowledgebase gene annotations and the set of annotations used in the raw experimental data using a more-generic set of location terms meant to rectify potential discrepancies in annotation granularity.

## HPA raw data-HPA knowledgebase data comparison using custom GOcats categories

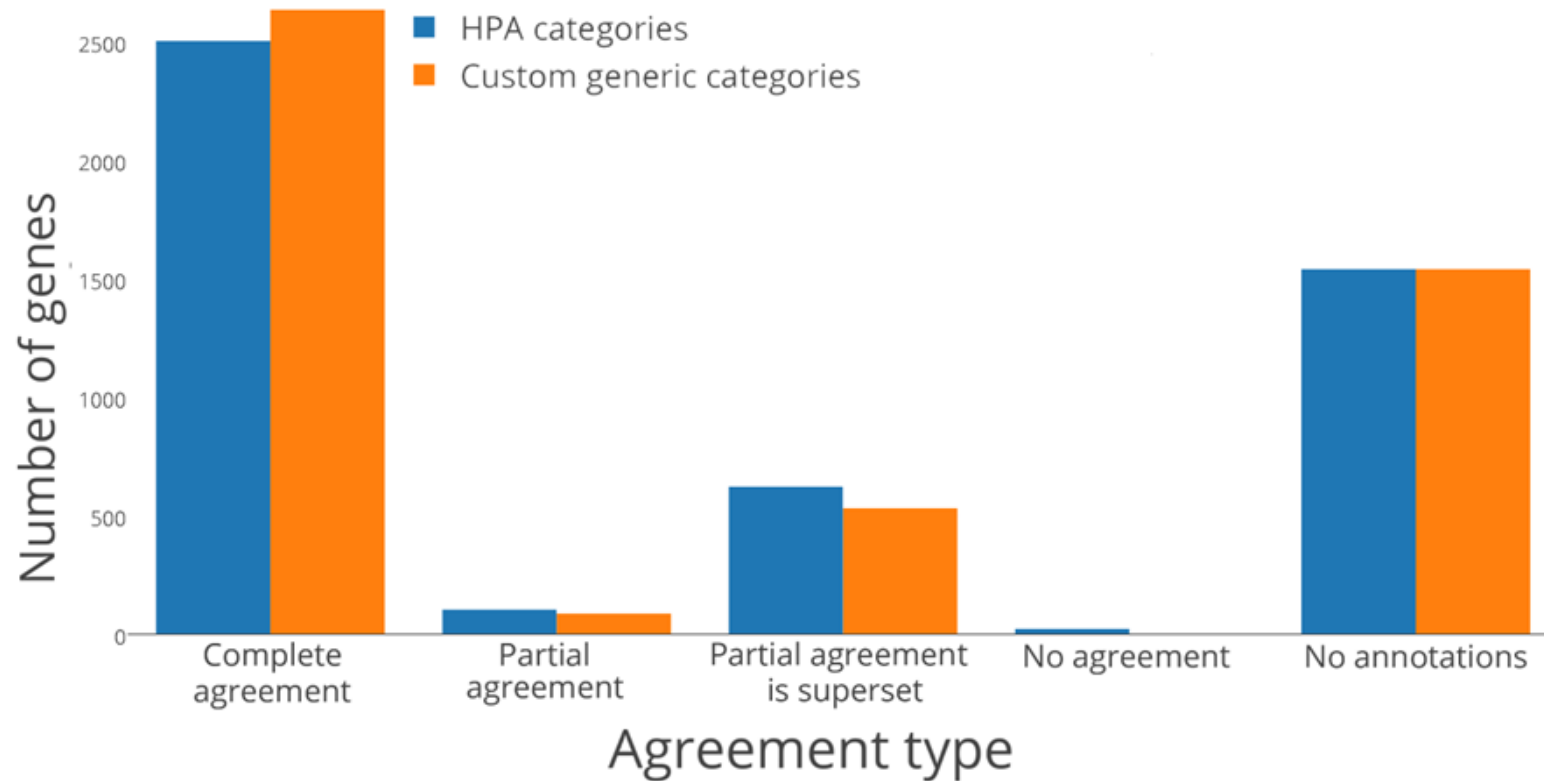


Figure 3.6 Comparison of HPA knowledgebase derived annotations to HPA experimental data

Figure 3.6 (continued) Number of genes in the given agreement type when comparing gene product annotations assigned by HPA in the EMBL-EBI knowledgebase to those in The Human Protein Atlas' raw experimental data. "Complete agreement" refers to genes where all subcellular locations derived from the knowledgebase and the HPA dataset matched. "partial agreement" refers to genes with at least one matching subcellular location. "partial agreement is superset" refers to genes where knowledgebase subcellular locations are a superset of the HPA dataset (these are mutually exclusive to the "partial agreement" category). "no agreement" refers to genes with no subcellular locations in common. "no annotations" refers to genes in the experimental dataset that were not found in the knowledgebase. The more-generic categories used in panel B can be found in Table 3.5

## CHAPTER 4. ADVANCES IN GENE ONTOLOGY UTILIZATION IMPROVE STATISTICAL POWER OF ANNOTATION ENRICHMENT

### 4.1 Background

Ontologies are used to document new knowledge gleaned from nearly every facet of biological and biomedical research today, and are created, maintained, and extended by experts with the goal of providing a unified annotation scheme that is readable by humans and machines (4). With the increased use of transcriptomics technologies, high-throughput investigation of the functional impact of gene expression in biological systems and disease processes via gene set enrichment analyses represents one important use of GO (10) (see Chapter 1.1).

While tools exist to incorporate GO annotations and the graphical structure of GO (i.e. ontological ancestor terms) in enrichment analyses, they fail to utilize the full extent of the semantic information available in GO due to limitations in how ontological relations are traversed. These limitations are due in part to the ambiguity in assessing which term, in a pair of terms connected by a relation is *more general* or *more specific* in the context of assigning the appropriate semantic *scope* while categorizing terminology. Details regarding this issue are provided in Chapter 1.8.

A separate, but related issue involves how some relations are defined and utilized in GO, leading to complications when drawing semantic inferences between terms. The relation in question here is *has\_part* which, contrary to intuition, is not a direct inverse of *part\_of* due to the concepts of universality. Details related to this are provided in Chapters 1.8 and 1.9. To summarize, in the context of inferring relations in a purely ontological

(linguistic) sense, the current axioms preventing the inversion of *has\_part* is important in preserving universal truths and thus avoiding illogical inferences such as “cell” *has\_part* “nucleus” therefore “nucleus” *part\_of* (implying every) “cell.” However, in the context of categorizing terminology for enrichment analyses, we argue that it is preferable to sacrifice perfect semantic accuracy in favor of increased information content by reinterpreting the inversed *has\_part* relation as *part\_of\_some*. In other word this relation would mean, “part of some, but not necessarily all.” The alternative solution, which other tools require, is that the *has\_part* relation is entirely dropped from the ontology. In our hypothetical example, this would mean that the connection between “cell” and “nucleus” would be lost altogether.

For the issues stated above, we have developed a new tool called the GO Categorization Suite (GOcats) (see Chapter 3). Fundamental to GOcats’ categorization algorithm is the re-evaluation of the *has\_part* edge as *part\_of\_some*—correcting semantic correspondence inferences while ensuring ubiquitous use of all categorization-relevant relations in GO.

For this investigation, the go-core version of the GO database was chosen in favor of the go-basic version, because it contains the *has\_part* edge relation which points away from the root of the ontology and because it contains other edges which connect the separate subontologies. Since one of our goals is to reinterpret mereological relations with respect to semantic scope, it is necessary that these relations be evaluated. Similarly, we excluded the go-plus version from this investigation, because we are not yet concerned with the reevaluation of the additional relations or database cross-references provided by go-plus.

While go-basic is a true DAG, go-core is not strictly acyclic due to the additional *has\_part* relations. However, when we inverse the traversal of *has\_part* into the *part\_of\_some* interpretation, acyclicity is maintained. Therefore, we refer to our modified go-core graph as a DAG. GOcats is a Python package written in version 3.4.2 of the Python program language (49). GOcats parses go-core and represents it as a DAG hierarchal structure. GOcats extracts subgraphs of the GO DAG (sub-DAGs) and identifies a representative node for each category in question (Figure 4.1). Details on GOcats' categorization algorithms can be found in sections 2.1.1 – 2.1.3. Full API documentation for GOcats is available online (53).

To overcome issues regarding scoping ambiguity among mereological relations, we hard-coded assigned properties indicating which term was broader in scope and which term was narrower in scope to each edge object created from each of the scope-relevant relations in GO. For example, in the node pair connected by a *part\_of* or *is\_a* edge, node 1 is narrower in scope than node 2. Conversely, node 1 is broader in scope than node 2 when connected by a *has\_part* edge (Table 4.1, Figure 4.2). This edge is therefore reinterpreted by GOcats as *part\_of\_some*. While the default scoping relations in GOcats are *is\_a*, *part\_of*, and *has\_part*, the user has the option to define the scoping relation set. For instance, one can create go-basic-like subgraphs from a go-core version ontology by limiting to only those relations contained in go-basic. For convenience, we have added a command line option, “go-basic-scoping,” which allows only nodes with *is\_a* and *part\_of* relations to be extracted from the graph.

In comparing GOcats' inclusion of re-evaluated *has\_part* relations to the traditional method of ignoring *has\_part* relations altogether and to the erroneous method of

misinterpreting native *has\_part* directionality, we illuminate the theoretical extent of information loss or potential for misinterpretation of *has\_part* relations, respectively. Furthermore, in two independent enrichment analyses of real data—from a publicly available breast cancer dataset (65) and from samples investigating equine cartilage development (66), we demonstrate that GOcats’ reinterpretation of *has\_part* can retain all information from GO while drawing appropriate categorical inferences in the context of annotation enrichment. Finally, we show that this reinterpretation has the added benefit of improving the statistical power of annotation enrichment analyses.

## 4.2 Results

### 4.2.1 GOcats’ Reinterpretation of the *has\_part* Relation Increases the Information Retrieval from GO and Avoids Potential Misinterpretations of Ambiguous Relationship Inferences

GOcats reevaluates path tracing for the *has\_part* edge to make it congruent with other relations that delineate scope. With path tracing unchanged, *has\_part* edges lead to erroneous term mappings unless they are completely excluded from the ontology. To evaluate the extent of incorrect semantic interpretation conferred by *has\_part* relations, we calculated all potential false mappings ( $pM_F$ ) between nodes for a given GO sub-ontology by counting the number of mappings from all children of a *has\_part* edge to all parents of a *has\_part* edge assuming the original GO *has\_part* edge directionality. Next, we compared the  $pM_F$  to the total number of true mappings ( $M_T$ ) for a given GO sub-ontology to evaluate the possible magnitude of their impact (Chapter subsections 4.5.1 and 4.5.2, Equations 1-5, (ref 63, SD1-2)). As shown in Table 4.2, there are 23,640  $pM_F$ s in Cellular

Component, 8,328 pM<sub>FS</sub> in Molecular Function, and 89,815 pM<sub>FS</sub> in Biological Process. Comparatively, the amount of pM<sub>FS</sub> is 42%, 13%, and 16% the size of the M<sub>T</sub> in Cellular Component, Molecular Function, and Biological Process, respectively.

The conventional solution to avoid these errors is to use versions of ontologies that remove edges like *has\_part*. (33). Considering the number of possible mappings between terms as a measure of information content, we quantified the loss of information acquired when *has\_part* is omitted during mapping by subtracting the number of M<sub>T</sub> in graphs containing *is\_a*, *part\_of*, and *has\_part* edges from those with only *is\_a* and *part\_of* edges. As shown in Table 4.2, Cellular Component lost 6,346 mappings, Molecular Function lost 6,242 mappings, and Biological Process lost 27,674 mappings, which equates to 11%, 10%, and 5% loss of information in these sub-ontologies, respectively. It is important to note that the mapping combinations were limited to those nodes containing *is\_a*, *part\_of*, and *has\_part* relations only. Because paths in GO are heterogeneous with respect to relation edges, this loss of information is a lower-bound estimate since other relations exist that connect additional nodes, but in a manner unusable for semantic correspondence interpretation. This is especially true for Biological Process, which has many regulatory relations that were not evaluated here.

While the potential for false mappings are high considering the *has\_part* relation alone, this statistic does not illuminate the scale of the issue facing users of current ontology mapping software. Importantly, it does not address a fundamental limitation and danger facing software like map2slim (M2S) (13), which non-discriminately evaluates relation edges. For example, terms linked by an active relation like *regulates*, or by the *has\_part* edge are categorized as if they are related by a scoping relation like *is\_a*. Therefore, we

calculated the total number of possible mappings produced by M2S and enumerated the intersection of these mappings against those made by GOcats which were constrained to paths that contained only scoping relations, *is\_a*, *part\_of*, and *has\_part* (Chapter 4.5.2, Equations 6 and 7). Overall, M2S made 325,180 GO term mappings, i.e. categorizations, which did not intersect GOcats' full set of corrected scoping relation mappings. We consider these false mapping pairs ( $M_{\text{pair},\text{M2S}}$ ), since they represent a problematic evaluation of scoping semantics. This contrasted with 710,961 correct mappings that intersected the GOcats mapping pairs ( $M_{\text{pair},\text{GOcats}}$ ) giving a percent error of 31.4%. Cellular Component, Molecular Function, and Biological Process contained 22,059, 29,955 and 273,166 erroneous mappings, which accounted for respective percent errors of 30.7%, 34.8%, and 31.1% (Table 3.3).

#### 4.2.2 GOcats' Reinterpretation of the *has\_part* Relations Provides Improved Annotation Enrichment Statistical Power

We incorporated GOcats-derived ontology ancestor paths (paths from fine-grained terms to more general, categorical terms) into the CategoryCompare version 1.99.158 (29) annotation enrichment analysis pipeline and performed annotation enrichment on an Affymetrix microarray dataset of ER+ breast cancer cells with and without estrogen exposure (65). We compared these enrichment results to those produced when unaltered ancestor paths from GO—excluding the *has\_part* relation—were incorporated into the same CategoryCompare pipeline (See Chapter 4.5.4 and (57)).

We also performed enrichment analyses comparing the ancestor traversals of DEseq2 differential gene expression datasets across time points during the fetal

development of two cartilage tissue types in *Equus caballus* in collaboration with Dr. James MacLeod and Dr. Rashmi Dubey (See Chapter subsections 2.2, 4.5.5, and 4.5.6, and (57)).

Assessment of adjusted p-values from significantly enriched terms using GOcats' paths versus the traditional method that omits *has\_part* edges shows that GOcats reliably improves the statistical significance of term enrichment results through its re-interpretation of *has\_part* relation semantics (Figure 4.3). In the breast cancer dataset, of the 217 significantly enriched terms found using the traditional enrichment method at an alpha of 0.01 for FDR-adjusted p-values, 182 had adjusted p-values that were improved when GOcats *part\_of\_some* paths were used. This number of improved p-values is statistically significant as indicated by a one-sided binomial test p-value of 1.86E-25 (i.e.  $1.86 \times 10^{-25}$ ). The full list of enriched terms and their adjusted p-values produced from GOcats' ancestor path tracing and *has\_part*-omitted ancestor path tracing for this analysis is provided in Supplemental Table 4.1.

Additionally, GOcats was able to identify 15 unique significantly-enriched terms at an alpha of 0.01 for adjusted p-values that would otherwise be omitted due to the loss of *has\_part* edges (Table 4.4). Four of these terms involve purinergic nucleotide receptor activity, which has been implicated elsewhere in other investigations related to breast cancer in both ER+ and ER- breast cancer cell lines.(67).

GOcats' path tracing showed similar improvements when comparing p-values from GO annotation enrichment derived from the differential gene expression analyses between equine cartilage development time points (Table 4.5). In this analysis (see Chapter 4.5.5), neighboring time point analyses (early and late) were compared to extreme time point

analyses (extreme) (Table 4.6). The traditional enrichment method yielded between 82 to 233 total enriched terms, with 67% to 92% of these terms' adjusted p-values being improved when GOcats ancestor path tracing was used. Quantifying the improvements in the p-values via a binomial test generates p-values ranging from 1.32E-03 to 2.58E-44 (i.e.  $1.32 \times 10^{-3}$  to  $2.58 \times 10^{-44}$ ). Even with a Bonferroni multiple test correction, the adjusted p-value of the six binomial tests performed range from 7.92E-03 and 1.55E-43.

Also, all but one of the binomial test p-values was below 6.22E-21; however, the comparison of the fetal interzone tissue at 45 days of gestation to neonatal epiphyseal cartilage had drastically fewer total enriched terms. Furthermore, GOcats was able to identify additional significantly-enriched terms from the first and second neighboring time point analyses as compared to the traditional method applied to the extreme analysis. GOcats extracts a notable number of uniquely enriched terms from the individual time point comparisons (Table 4.6, UniqueEnrichedTerms<sub>GOcats</sub>). A few of these enriched terms (Table 4.6, SupportedEnrichedTerms) are directly supported by the traditional method enrichment of the extreme time point comparisons. In other words, the traditional method's enrichment of the extreme time point comparisons provides some ground truth for validating uniquely enriched terms detected by the GOcats enrichment analysis of the nearest-neighbor time point comparisons.

## 4.3 Discussion

### 4.3.1 Issues with Semantic Correspondence

As early as the late 1980s, explicit definitions of semantic correspondence for a relation between ontological terms have been stressed in the context of relational database

design (68). This includes concepts of part-whole (mereology), general-specific (hyponymy), feature-event, time-space (i.e spatiotemporal relations), and others. OBO's and GO's ontological edges are directional insofar as their relations accurately describe how the first node relates to the second node empirically, providing axioms for deriving direct semantic inferences. However, the directionality of these edges is ambiguous in that they do not explicitly describe how the terms relate to one another semantically in terms of scope, and this is due largely to the lack of explicit semantic correspondence qualifiers.

A simple way to avoid mapping problems associated with non-scoping relation direction is to omit those relations from the analysis. This strategy avoids incorrect scoping interpretation at the expense of losing information. As an example, EMBL-EBI's QuickGO term mapping service omits *has\_part* type under its "filter annotations" by GO identifier options (33). Furthermore, Bioconductor's GO.db (69) also avoids mapping issues by indirectly omitting this relation; it uses a legacy MySQL dump version of GO which does not contain relation tables for *has\_part*. We argue that while avoiding problematic relations altogether does prevent scope-specific mapping errors, it also limits the amount of information that can be gleaned from the ontology. By eliminating *has\_part* from graphs created by GOcats, we see a ~11% decrease in information content (as indicated by a decrease in the number possible mappings) in Cellular Component. Likewise, there is a 10% and 5% decrease of information content in Molecular Function and Biological Process, respectively (Table 4.2). Thus, omitting these relations from analyses removes a non-trivial amount of information that could be available for better interpretation of functional enrichment. However, the total impact is not completely appreciated here, because not all relations were evaluated in this study; only the scoping relations of *is\_a*,

*part\_of*, and *has\_part*. The potential for additional information loss is very high in Biological Process, for example, when considering the large number of unaccounted relations *regulates*, *positively\_regulates*, and *negatively\_regulates* (Table 4.1). These relations add critical additional regulatory information to ontological graph paths, which would also be lost when ignoring the *has\_part* relation, if they occurred along a path that also contained *has\_part*. The same is also true for Molecular Function, although the frequency of additional, non-scoping relations are lower.

Furthermore, automated summarization of annotations enriched in gene sets requires a more sophisticated evaluation of the scoping semantics contained in ontologies, which prior tools are not fully equipped to provide. M2S is one widely-utilized GO term categorization method that is available as part of the OWLTools Java application. The Perl version of M2S has been integrated into the Blast2GO suite since 2008 (70) and this gene function annotation tool has been cited in over 1500 peer-reviewed research articles (Google Scholar as of Nov. 28, 2017). We verified that the Perl and Java versions of M2S produced identical GO term mappings for a given dataset and GO slim, and therefore have the same mapping errors (ref 63, SD2). Although the number of pM<sub>FS</sub> reported in the results represent the upper limit of the possible erroneous mappings, the fact that at least 120,000 of these exist in GO for the *has\_part* relation alone or that the removal of this edge type results in up to an 11% reduction of information content provide bounds on the scope of the issue. To be clear, tools like M2S can be safe and not produce flawed mappings if they are used alongside ontologies that contain only those relations that are appropriate for evaluation, such as *go-basic*. However, we intentionally utilized *go-core* to illustrate the

danger in using tools that do not provide explicit semantic control on how ontologies are utilized.

GOcats represents a step toward a more thorough evaluation of the semantics contained within ontologies by handling relations differently according to the type of correspondence that they represent. In the case of relations such as *has\_part*, this involves altering the correspondence directionality for the task at hand, which is to organize terms into categories. As a proof-of-concept, we classified the *is\_a*, *has\_part*, and *part\_of* relations into a common “scoping” correspondence type and hard-coded assigned graph path tracing heuristics to ensure that they are all followed from the narrower-scope term to the broader-scope term. One caveat of this approach is that because of previously mentioned issues in universality logic, the inverse of *has\_part* is not strictly *part\_of*, but rather *part\_of\_some*. We argue that the highly unlikely misinterpretation of universality in this strategy is preferable to the loss of information experienced when using trimmed versions of ontologies for term categorization. To elaborate, most current situations calling for term categorization involve gene enrichment analyses. Spurious incorrect mappings through *part\_of\_some* edges would not enrich to statistical significance, unless a systematic error or bias is present in the annotations. Even if a hypothetical term categorization resulted in enrichment of a general concept that was not relevant to the system in question (i.e. “nucleus” enriched in a prokaryotic system), it would be relatively straight-forward to reject such an assignment by manual curation and find the next most relevant term. Conversely, it is not reasonable to manually curate all possible missed term mappings resulting from the absence of an edge type in the ontology.

Another potential complication in semantic correspondence of relations is that some relations are *inherently* ambiguous. The clearest example of this again can be found in the well-utilized *part\_of* relation. This relation is used to describe relations between physical entities and concepts (e.g. “nuclear envelope” *part\_of* “endomembrane system”) and between two concepts (e.g. “exit from mitosis” *part\_of* “mitotic nuclear division”) with no explicit distinction. To address the former issue, future work will augment our use of hard-coded categorization of semantic correspondences through the development of heuristic methods that identify and categorize these among the hundreds of relations in the Relations Ontology (1) (71). As a good starting point, we suggest using five general categories of relational correspondence for reducing ambiguity (Table 4.1): scope (hyponym-hypernym), mereological, a subclass of scope (meronym-holonym), spatiotemporal (process-process, process-entity, entity-entity), active (actor-subject), and other.

#### 4.3.2 Using GOcats for Annotation Enrichment

While we reported the loss of information available for annotation enrichment with *has\_part* excluded from GO and quantified the effect of incorrect inferences that can be made if *has\_part* is included in GO during enrichment, these results only represent hypothetical effects that might be overcome when GOcats reinterprets this relation. One of GOcats’ original intended purposes was to improve the interpretation of results from annotation enrichment analyses. However, in the process of designing heuristics to appropriately categorize GO terminology, we also sought to overcome the limitations that come with following the traditional methods of path tracing along relations in GO. Here

we focused on overcoming the loss of information encountered when ignoring *has\_part* relations. Our solution was to re-evaluate these relations under the logic of *part\_of\_some* and invert the direction of *has\_part*. While this re-interpretation is limited in usage, we believe that, in the scope of annotation enrichment, it is valid for reasons previously explained.

Our first evaluation of enrichment results compared GOcats' ancestor paths to traditional GO ancestor paths in the enrichment analysis of an older, publicly-available microarray breast cancer dataset, generated from an Affymetric HG-U95Av2 array which only covered 9000 genes. With this comparison, we demonstrate a highly statistically significant improvement ( $p\text{-value}=1.86\text{E-}25$ ) in the statistical power of annotation enrichment analysis. Specifically, 182 out of 217 significantly enriched GO terms from the traditional analysis had improved p-values in the GOcats-enhance enrichment analysis. Importantly, we also detect significantly enriched GO terms in the GOcats' results that were not detected using the traditional analysis. The inclusion of the re-interpretation of *has\_part* edges allowed for the significant enrichment (adjusted-p-value < 0.002 with FDR set to 0.01) of four terms related to purinergic nucleotide receptor signaling which has been associated with ER+ MCF-7 breast cancer cell proliferation (72,73). Furthermore, purinergic nucleotide receptor signaling has been implicated in predicting breast cancer metastasis in other studies; however, these studies involved ER- metastatic breast cancer cell lines (74). We again confirmed this effect in our evaluation of GO annotation enrichment results of recently collected RNAseq equine cartilage development datasets. Here we saw an improvement in 67% to 92% of enriched terms across the six time point enrichment analyses. Fundamentally, the addition of *part\_of\_some* interpretation of

*has\_part* relations improves the statistical power of the annotation enrichment analysis, allowing the detection of additional enriched annotations with statistical significance from the same dataset. In addition, the GOcats annotation enrichment analysis extracts a notable number of uniquely enriched annotations from the neighboring, individual time point differential gene expression analyses. Some of these uniquely enriched terms are directly supported by the traditional annotation enrichment analysis of the extreme time point differential gene expression analyses (Table 4.6). These results on multiple datasets involving two separate experimental designs using both older and more recent transcriptomics technologies demonstrate the robustness of utilizing GOcats-augmented ontology paths to derive additional information from annotation enrichment analyses. While these results demonstrate an improvement in statistical power of annotation enrichment analysis, no data analysis method can address unknown bias in a dataset. Bias that leads to confounding factors is best addressed at the point of experimental design, but sometimes the effects from identified confounding factors can be mitigated after the experiment during data analysis (75).

#### 4.4 Conclusions

To conclude, GOcats enables the simultaneous extraction and categorization of gene and gene product annotations from GO-utilizing knowledgebases in a manner that respects the semantic scope of relations between GO terms. It also allows the end-user to organize ontologies into user-defined biologically-meaningful concepts—a feature that we have explained elsewhere (76). This categorization lowers the bar for extracting useful information from exponentially growing scientific knowledgebases and repositories in a semantically safer manner. In summary, GOcats is a versatile software tool applicable to

data mining, annotation enrichment analyses, ontology quality control, and knowledgebase-level evaluation and curation.

## 4.5 Methods

### 4.5.1 Evaluating Hypothetical False Mapping and True Mapping Pairs in GO Involving the *has\_part* Relation

To determine how significant mapping issues are because of semantic scope inconsistencies with *has\_part* relations, we built the GO graph, data-version: releases/2016-01-12 using only the scoping relations *is\_a*, *part\_of*, and *has\_part* edges, while omitting other relation edges in the graph, such as *regulates*, *happens\_during*, and *ends\_during*. Next, we counted the number of potential false mappings (pMF) that could result if *has\_part* was left in its unaltered directionality; i.e. the edge directionality that currently exists in GO. To accomplish this, we define sets of potentially problematic ancestors ( $PA_e$ ) for every *has\_part* edge ( $e$ ) as

$$PA_e = \{Ae_{child} + e_{child}\} - \{Ae_{par} + e_{par}\} \quad (1)$$

where  $Ae_{child}$  and  $Ae_{par}$  are sets of nodes that are ancestors of the edge's child and parent nodes, respectively, and  $e_{child}$  and  $e_{par}$  are the edge's parent and child nodes. Similarly, we define the potentially problematic descendants ( $PDe$ ) for every *has\_part* edge ( $e$ ) as

$$PDe = \{De_{par} + e_{par}\} - \{De_{child} + e_{child}\} \quad (2)$$

where  $De_{par}$  and  $De_{child}$  are sets of nodes that are descendants of the edge's parent and child nodes, respectively. We then calculate the potential mappings that can occur across each edge,  $e$  by the following:

$$pM_{F,e} = \{(d, a) \mid d \in PD_e; a \in PA_e\} \quad (3)$$

The total number of potential false mappings that can result from an edge type, in this case the *has\_part* relation, is given by

$$pM_F = \left| \bigcup_{e=1}^n pM_{F,e} \right| \quad (4)$$

Finally, we calculate the number of total possible true mappings (MT) between any two arbitrary nodes ( $n_1, n_2$ ) in a given sub-ontology graph (G) in GO:

$$M_T = |\{n_1 \text{anc} \cap n_2 \text{desc} \mid n_1 \in G; n_2 \in G\}| \quad (5)$$

In Equation 6, we used GOcats to calculate the possible number of true mappings while considering *is\_a*, *part\_of*, and re-evaluated *has\_part* (*part\_of\_some*) relations in GO.

#### 4.5.2 Evaluating Hypothetical False Mappings Encountered When the Unaltered *has\_part* Relation is Parsed with Map2Slim

The Java implementation of OWLTools' Map2Slim (M2S) does not include the ability to output a mapping file between fine-grained GO terms and their GO slim mapping target from the GAF that is mapped. To identify target ancestor terms of individual GO terms, we created a special custom GAF where the gene ID column and GO term annotation column of each line were each replaced by a different GO term for each GO term in Cellular Component, data-version: releases/2016-01-12. We then allowed M2S to map this GAF with a provided GO slim. The resulting mapped GAF was parsed to create a standalone mapping between the terms from the GO slim and a set of the terms in their subgraphs. Because M2S's custom term list option removes terms subsumed by other mappings, we were forced to also perform separate mappings for each GO term; e.g. the entire GO was

mapped to one GO term at a time for each ~44,000 terms. These computations were done in parallel on a small TORQUE-managed Linux cluster to complete the calculations in a reasonable amount of time. We combined and converted the results into a set of ordered term pairs ( $M_{pair,M2S}$ ), where the first position is the mapped term and the second position is the term to which the first is mapped; self-mappings were ignored. Using the GOcats' evaluation of the three scoping relations, *is\_a*, *part\_of*, and *has\_part*, to create the “correct” set of mappings in a scoping paradigm, we defined the set of potentially false M2S mappings ( $pM_{f,M2S}$ ) as

$$pM_{f,M2S} = \{M_{pair,M2S}\} - (\{M_{pair,M2S}\} \cap \{M_{pair,GOcats(scoping)}\}) \quad (6)$$

where  $M_{pair,GOcats(scoping)}$  is the set of ordered GO term mapping pairs produced from GOcats, under the constraint that only scoping relations were used in the graph (*is\_a*, *has\_part*, and *part\_of*). The ratio of potential false scoping-type mappings to correct scoping mappings produced by M2S ( $M2S_{error}$ ) is given by

$$M2S_{error} = \frac{|pM_{f,M2S}|}{|\{M_{pair,GOcats(scoping)}\}|} \quad (7)$$

To look specifically at individual sub-ontologies, we filtered the M2S mapping pairs to those where both terms were a member of each sub-ontology. These were also intersected with the full set of GOcats mapping pairs (ref 63, SD1).

#### 4.5.3 Comparing Mapping Functionality between the Java and Perl Versions of Map2Slim

To ensure that the same mapping errors encountered using the Java version of M2S, which is integrated in OWLTools, are also present in the Perl version of M2S, which is

integrated in Blast2GO, we tested whether the mapping functionality was consistent between the two versions. Since the Perl version only supports GO slims and does not support custom specification of a list of GO terms, we compared the output of each version’s mapping of the HPA-sourced knowledge data to the “generic” GO slim dataset (32). Since some minor GAF formatting differences exist between the output files, we wrote a script to directly compare the gene-to-GO annotation mappings made by each version (ref 63, SD2).

#### 4.5.4 Annotation Enrichment Analysis of Breast Cancer Dataset

To evaluate the effects that GOcats ancestor paths had on real data, we performed GO annotation enrichment using `categoryCompare` (29)—and an updated version of the GO graph, data-version: releases/2017-12-02—on an Affymetrix microarray dataset of ER+ breast cancer cells with and without estrogen exposure (65). In this dataset, we ignored time point information and only considered data associated with the presence and absence of estrogen exposure.

The `categoryCompare` package can consider GO ancestor terms for annotated terms in the experimental dataset when calculating enrichment. We therefore created two mapping dictionaries in Python where a key of each term in GO maps to a set of its ancestor terms in the GO graph. For the traditional method of inferring ancestors, we created this mapping from a version of the GO graph with the *has\_part* relation omitted. For testing GOcats’ effect on enrichment, we created a version of this mapping with the *has\_part* relation re-interpreted as *part\_of\_some*. We applied these ancestor mappings to all

annotations in the human GOA database, generated: 2017-11-21 08:07 (77). R scripts and Python scripts for generating the enrichment results can be found in (ref 63, SD3).

To compare FDR-adjusted (target FDR=0.01) p-values between enrichment results produced by GOcats ancestors and traditional ancestors, we filtered the enriched terms identified by the traditional method with an alpha cutoff of 0.01 and counted the number of terms identified by GOcats' analysis whose adjusted p-value was less than the traditional analysis. Identical adjusted p-values were ignored. We then performed a one-sided binomial test (i.e. "coin-toss analysis" with directional change from 0.5) comparing the number of significantly enriched adjusted p-values that improved with GOcats versus total number of enriched terms found in the traditional analysis (with identical adjusted p-values excluded). To identify uniquely enriched terms found using the GOcats-enhanced enrichment analysis, we compared the sets of significantly enriched terms (alpha cutoff 0.01 for adjusted p-values) in each enrichment results table and selected terms only found in the GOcats-enhanced set.

#### 4.5.5 Annotation Enrichment of Equine Cartilage Development Dataset

To further test the effects that GOcats' ancestor path tracing has on term enrichment, we again performed GO annotation enrichment using categoryCompare (29) applied to differentially-expressed genes identified by DESeq2 from RNAseq datasets derived from developing equine cartilaginous tissues (interzone and anlagen) across two gestational time points and their neonatal derivatives (articular cartilage and epiphyseal cartilage, respectively). The time points were fetal interzone tissue at 45 days of gestation (iz\_45); fetal anlagen tissue at 45 days (anl\_45); fetal interzone tissue at 60 days of

gestation (iz\_60); anlagen fetal tissue at 60 days (anl\_60); neonatal articular cartilage (ac\_neo); and neonatal epiphyseal cartilage (epi\_neo). At least six biological replicates were acquired for each tissue type and time point (separate equine fetuses from similar breeds) with RNA-seq readings of 30-40 million reads per sample.

We downloaded equine gene annotations from AgBase (78) and built two full ancestor annotation mappings for each gene, one using GOcats' re-evaluation of the *has\_part* relation and the other using the traditional method of omitting the *has\_part* relation altogether.

For each pairwise time point comparison from the DESeq2 analyses (iz/anl\_45-iz/anl\_60, iz/anl\_60-ac/Epi\_neo, or iz/anl\_45-ac/Epi\_neo), we selected positively- or negatively-changing genes by filtering to those changing genes which had an adjusted p-value  $\leq 0.01$ . Based on the sign of each gene's fold expression from the dataset we classified these genes into categories for categoryCompare as "positive", "negative", or "all" (either positively or negatively changing in expression). Enrichment was performed on each of these three categories for each three pairwise time point comparisons (early, late, and extreme) for each two tissue types using two ancestor mappings: GOcats' and the traditional omission of *has\_part*, yielding 36 total enrichment analyses.

Using the enrichment results from the "all" category for each pairwise time point comparison and tissue type, we again evaluated the improvement in the adjusted p-value seen using the GOcats' ancestors when compared to the traditional method of mapping ancestors using a binomial test (see Chapter 2.2 for details).

In addition to the “positive”, “negative”, and “all” gene sets identified from the individual pairwise time point analyses, we also defined special gene sets relating to the scope of the whole time series. These were defined as i) early: those genes that significantly increased or decreased in fold-change during the iz/anl\_45-iz/anl\_60 time point comparison but did not significantly change in the iz/anl\_60-ac/epi\_neo time point comparison ii) late: those genes that did not have a significant fold-change in the iz/anl\_45-iz/anl\_60 time point comparison but did significantly change in the iz/anl\_60-ac/epi\_neo time point comparison iii) transient: those genes that significantly change during the iz/anl\_45-iz/anl\_60 time point comparison but then significantly change in the opposite direction during the iz/anl\_60-ac/epi\_neo time point comparison and iv) consistent: those genes that experience fold change in expression consistently throughout the time series. We also divided each of these whole time series gene sets into positive and negative sets corresponding to the sign of the fold-change. In the case of transient, the directionality corresponds to the fold change in the first, iz/anl\_45-iz/anl\_60 time point comparison.

To evaluate GOcats’ potential to improve the statistical power of annotation enrichment, we compared early and late time point annotation enrichments derived from GOcats ancestor traversal to the extreme time points annotation enrichment derived from traditional ancestor traversal. Here we define the following sets of annotations for each tissue type evaluated:

$$\text{EarlyUniqueEnrichedTerms}_{\text{Gocats}} = 45\_to\_60_{\text{Gocats}} - 45\_to\_60_{\text{no\_hp}} - \text{Transient}_{\text{no\_hp}} \quad (8)$$

The  $45\_to\_60_{\text{Gocats}}$  and  $45\_to\_60_{\text{no\_hp}}$  variables are the sets of GO terms identified when comparing the iz/anl\_45 time point to the iz/anl\_60 time point using GOcats or the traditional ancestor mapping method of ignoring the *has\_part* relation, respectively.

$\text{Transient}_{\text{no\_hp}}$  is the set of enriched terms categorized as transient for the whole time series using the traditional ancestor mapping method.

$$\text{EarlySupportedEnrichedTerms} = \text{EarlyEnrichedTerms}_{\text{GOcats}} \cap \text{Consistent}_{\text{no\_hp}} \quad (9)$$

$\text{Consistent}_{\text{no\_hp}}$  is the set of enriched terms categorized as consistent for the whole time series using the traditional ancestor mapping method.

$$\text{LateUniqueEnrichedTerms}_{\text{GOcats}} = 60\_to\_neo_{\text{GOcats}} - 60\_to\_neo_{\text{no\_hp}} - \text{Transient}_{\text{no\_hp}} \quad (10)$$

The  $60\_to\_neo_{\text{GOcats}}$  and  $60\_to\_neo_{\text{no\_hp}}$  variables are the sets of GO terms identified when comparing the *iz/anl\_60* time point to the *ac/api\_neo* time point using GOcats or the traditional method of ignoring the *has\_part* relation, respectively.

$$\text{LateSupportedEnrichedTerms} = \text{LateEnrichedTerms}_{\text{GOcats}} \cap \text{Consistent}_{\text{no\_hp}} \quad (11)$$

#### 4.5.6 RNASeq Analysis of Equine Cartilage Development Time Points.

Our collaborators, Dr. James MacLeod and Dr. Rashmi Dubey, collected tissue samples across six experimental groups (Table 4.7) and compared differential gene expression at a transcriptome level using mRNA sequencing. The following protocol was executed by these collaborators. Sample collection methods have been described previously (66,79) and were conducted in accordance with an approved University of Kentucky Institutional Animal Care and Use Committee protocol (# 2014-1215). Total RNA was isolated using a commercial kit (Qiagen RNeasy Micro Kit, cat# 74004) after homogenization on ice as previously described (80). Following ethanol precipitation and re-solubilization in sterile distilled water, the total RNA was quantified using a fluorometric assay (Qubit, Life Technologies, Q10210, Q32852) and assessed for chemical

contaminants using a spectrophotometer (NanoDrop ND 1000) and for structural integrity with a Bioanalyzer 2100 (Agilent Technologies, Eukaryotic Total RNA Nano & Pico Series II). All RNA samples met quality thresholds of 260/280 absorbance ratios of 1.7-2.0, 260/230 absorbance ratios of 1.8-2.1, and an Agilent RNA integrity number (RIN) of  $\geq 7.0$ .

RNAseq libraries were constructed using the TruSeq HT Stranded RNA Sample Preparation Kit (Illumina San Diego, CA). PolyA<sup>+</sup> RNA was selected from 1  $\mu$ g of total RNA and first-strand synthesis performed using random hexamer primers and SuperScript II<sup>TM</sup> reverse transcriptase (Life Technologies). Resulting double-stranded cDNA was then blunt-ended and ligated to indexed adaptors, followed by PCR amplification for 12 cycles with Kapa HiFi polymerase (Kapa Biosystems, Woburn, MA). Libraries were initially quantitated using Quant-it<sup>©</sup> (Life Technologies, Grand Island, NY) and the average size determined on an AATI Fragment Analyzer (Advanced Analytics, Ames, IA). They were then diluted to a final concentration of 5nM and further quantitated by qPCR on a BioRad CFX Connect Real-Time System (Bio-Rad Laboratories, Inc. CA).

Strand-specific sequencing was performed using a paired-end mRNA-seq protocol at the Roy J. Carver Biotechnology Center, University of Illinois at Urbana-Champaign. A minimum of 30 million reads were generated for each sample, trimmed (Trimmomatic Version 0.36 (81)), and then mapped to the equine reference genome (EquCab2.0, chromosomes 1-31, M, X, and Un, NCBI Annotation Release 102) using MapSplice 3.0 Beta (82). Default settings were used. Steady state levels of mRNA levels were compared between the six experimental groups at all protein-coding gene loci structurally annotated in the equine genome (EquCab2.0, NCBI Annotation Release 102) by DESeq2 analysis

(26). DESeq2 modeled the read count data using negative binomial distribution and performed the statistical testing for differential gene expression. The analysis returned a p-value determined by Wald statistics and an adjusted p-value (to apply corrections for multiple comparisons testing). The Benjamini-Hochberg multiple-test correction was applied to evaluate the false-discovery rate (FDR). The DESeq2 identified 5572 (ANL\_45 to ANL\_60), 5464 (ANL\_45 to Epi\_neo), 7049 (ANL\_60 to Epi\_neo), 9929 (IZ\_45 to IZ\_60), 9975 (IZ\_45 to AC\_neo), and 8329 (IZ\_60 to AC\_neo) differentially expressed genes, which have an adjusted p-value  $< 0.01$  after multiple testing corrections.

Scripts and snakemake (54) workflows for performing annotation enrichment across genes identified from the results of these DeSEQ2 analyses can be found in our FigShare directory (ref 63, SD4).

Table 4.1 Frequency of Relations in the Gene Ontology and Suggested Semantic Correspondence Classes to Reduce Ambiguity.†

Relationship	Frequency in GO (CC+BP+MF)	Frequency in GO CC	Frequency in GO BP	Frequency in GO MF	Correspondence Class	Correspondence Members
is_a	72455	5591	54689	12175	Scoping (hyponymy)	hyponym "is_a" hypernym
part_of	8613	1702	5751	1160	Scaling (meronymy)	meronym "part_of" holonym
has_part	736	156	339	241	Scaling (meronymy)	holonym "has_part" meronym
happens_during	24	0	24	0	Spatiotemporal (process-process)	process "happens_during" process
ends_during	1	0	1	0	Spatiotemporal (process-process)	process "ends_during" process
occurs_in	181	0	180	1	Spatiotemporal (process-entity or process-process)	process "occurs_in" entity OR process "occurs_in" process
regulates	3368	0	3322	46	Active (actor-subject)	actor "regulates" subject
positively_regulates	2916	0	2880	36	Active (actor-subject)	actor "positively_regulates" subject
negatively_regulates	2937	0	2285	52	Active (actor-subject)	actor "negatively_regulates" subject
regulated_by‡	0	0	0	0	Active (actor-subject)	subject "regulated_by" actor
before‡	0	0	0	0	Spatiotemporal (prior-latter)	prior "before" latter

† GO-core data-version: releases/2016-01-12 (available in (57))

‡ These relationships are not found in GO but are part of the Relations Ontology

Table 4.2 Prevalence of Potential *has part* Relation Mapping Errors in GO.

<b>Sub-Ontology</b>	<b>Estimated Potential False Mappings (epM<sub>F</sub>)</b>	<b>True Mappings (M<sub>T</sub>)</b>	<b>M<sub>T</sub> ∩ epM<sub>F</sub></b>	<b>Potential False Mappings pM<sub>F</sub> = epM<sub>F</sub> (M<sub>T</sub> ∩ epM<sub>F</sub>)</b>	<b>True Mappings without HP (IA_PO M<sub>T</sub>)*</b>	<b>Lost Mappings (M<sub>T</sub> - IA_PO M<sub>T</sub>)*</b>
Cellular Component	30036	56025	6396	23640	49679	6346
Molecular Function	10074	62436	1746	8328	56194	6242
Biological Process	93092	555543	3277	89815	527869	27674

\* IA\_PO refers to a graph created with only is\_a and part\_of relationship edges.

Table 4.3 Summary of GO term Mapping Errors Resulting from Misevaluation of Relations with Respect to Semantic Scoping

<b>(Sub) Ontology</b>	<b>Map2Slim Mappings (<math>M_{\text{pair},M2S\_ont}</math>)*</b>	<b>GOcats Scoping Mappings (<math>M_{\text{pair},Gocats\_ont}</math>)*</b>	<b>Potentially false Map2Slim Mappings <math>pM_{F,M2S} = M_{\text{pair},M2S} - (M_{\text{pair},M2S} \cap M_{\text{pair},Gocats\_all})^*</math></b>	<b>Map2Slim Correct Mappings <math>M_{T,M2S} = M_{\text{pair},M2S} \cap M_{\text{pair},Gocats\_all}^*</math></b>	<b>Possible Map2Slim Error Fraction <math>pM_{F,M2S} / M_{\text{pair},M2S\_ont}</math></b>
All GO	1036141	820467	325180	710961	0.314
Cellular Component	71835	56025	22059	49776	0.307
Molecular Function	86163	62436	29955	56208	0.348
Biological Process	878143	555543	273166	604977	0.311

\* GOcats\_all refers to GOcats-derived mapping pairs across all of GO, while GOcats\_ont refers to GOcats-derived mapping pairs for the indicated ontology in each row.

Table 4.4 Uniquely Enriched Terms between GOcats Paths and Traditional Paths from the Breast Cancer Dataset Analysis

GO Term	Description	Adjusted p-value	Uniquely enriched in
GO:0035590	purinergic nucleotide receptor signaling pathway	0.000119296	GOcats
GO:0016502	nucleotide receptor activity	0.000103448	GOcats
GO:0035586	purinergic receptor activity	0.000129432	GOcats
GO:0036387	pre-replicative complex	6.03E-05	GOcats
GO:0042023	DNA endoreduplication	2.70E-10	GOcats
GO:0006313	transposition, DNA-mediated	1.31E-28	GOcats
GO:0031261	DNA replication preinitiation complex	5.55E-06	GOcats
GO:0032196	transposition	1.31E-28	GOcats
GO:0004888	transmembrane signaling receptor activity	0.006197782	GOcats
GO:0035587	purinergic receptor signaling pathway	0.000129432	GOcats
GO:0098039	replicative transposition, DNA-mediated	1.31E-28	GOcats
GO:0099600	transmembrane receptor activity	0.006197782	GOcats
GO:0001614	purinergic nucleotide receptor activity	0.000119296	GOcats
GO:0005656	nuclear pre-replicative complex	6.03E-05	GOcats

GO:0000988	transcription factor activity, protein binding	0.002944403	GOcats
GO:0051716	cellular response to stimulus	0.008043537	Traditional paths
GO:0007059	chromosome segregation	1.54E-06	Traditional paths
GO:0045005	DNA-dependent DNA replication maintenance of fidelity	0.001514676	Traditional paths
GO:0008094	DNA-dependent ATPase activity	0.000454406	Traditional paths
GO:0140097	catalytic activity, acting on DNA	6.04E-09	Traditional paths
GO:0050896	response to stimulus	0.000712619	Traditional paths
GO:1902969	mitotic DNA replication	0.001852706	Traditional paths

Table 4.5 Binomial Test Results for GOcats Verses Traditional Enrichment for Equine Cartilage Development Time Point Comparisons

<b>Tissue Type</b>	<b>Time Series Comparison</b>	<b>Total Enriched Terms</b>	<b>Enriched Terms with Lower P-value with GOcats*</b>	<b>One-sided Binomial Test</b>
Anlagen	45-day fetal to 60-day fetal (early)	228	183	6.22E-21
	60-day fetal to neonatal (late)	140	129	5.31E-27
	45-day fetal to neonatal (extreme)	158	139	5.01E-24
Interzone	45-day fetal to 60-day fetal (early)	82	55	1.32E-03
	60 day fetal to neonatal (late)	233	196	1.23E-27
	45-day fetal to neonatal (extreme)	233	215	2.58E-44

Table 4.6 Neighbor Versus Extreme Time Point Comparison of Enriched Terms in Equine Cartilage Development Enrichment Analyses

Tissue type	GO Term Set	Terms in set
anlagen	EarlyEnrichedTerms	50
	EarlySupportedEnrichedTerms <sup>‡</sup>	1
	EarlyUniqueEnrichedTerms <sub>Gocats</sub> <sup>‡</sup>	49
	LateEnrichedTerms	41
	LateSupportedEnrichedTerms <sup>‡</sup>	0
	LateUniqueEnrichedTerms <sub>Gocats</sub> <sup>‡</sup>	41
Interzone	EarlyEnrichedTerms	22
	EarlySupportedEnrichedTerms <sup>‡</sup>	3
	EarlyUniqueEnrichedTerms <sub>Gocats</sub> <sup>‡</sup>	19
	LateEnrichedTerms	81
	LateSupportedEnrichedTerms <sup>‡</sup>	3
	LateUniqueEnrichedTerms <sub>Gocats</sub> <sup>‡</sup>	78

<sup>‡</sup> Sets defined in equations 8-11

Table 4.7 Comparison of Equine fetus tissue samples

Sample Description		Age	Tissue source
Equine Fetus	Interzone (n=7)	45-46 days gestation	Carpal and tarsal joints
	Anlage (n=6)		Metaphysis of distal humerus and femur
Equine Fetus	Interzone (n=7)	57-66 days gestation	Carpal joints
	Anlage (n=7)		Metaphysis of distal humerus and femur
Equine Neonate	Articular cartilage (n=7)	0-9 days postnatal	Femorotibial joint
	Epiphyseal cartilage (n=7)		Proximal tibia

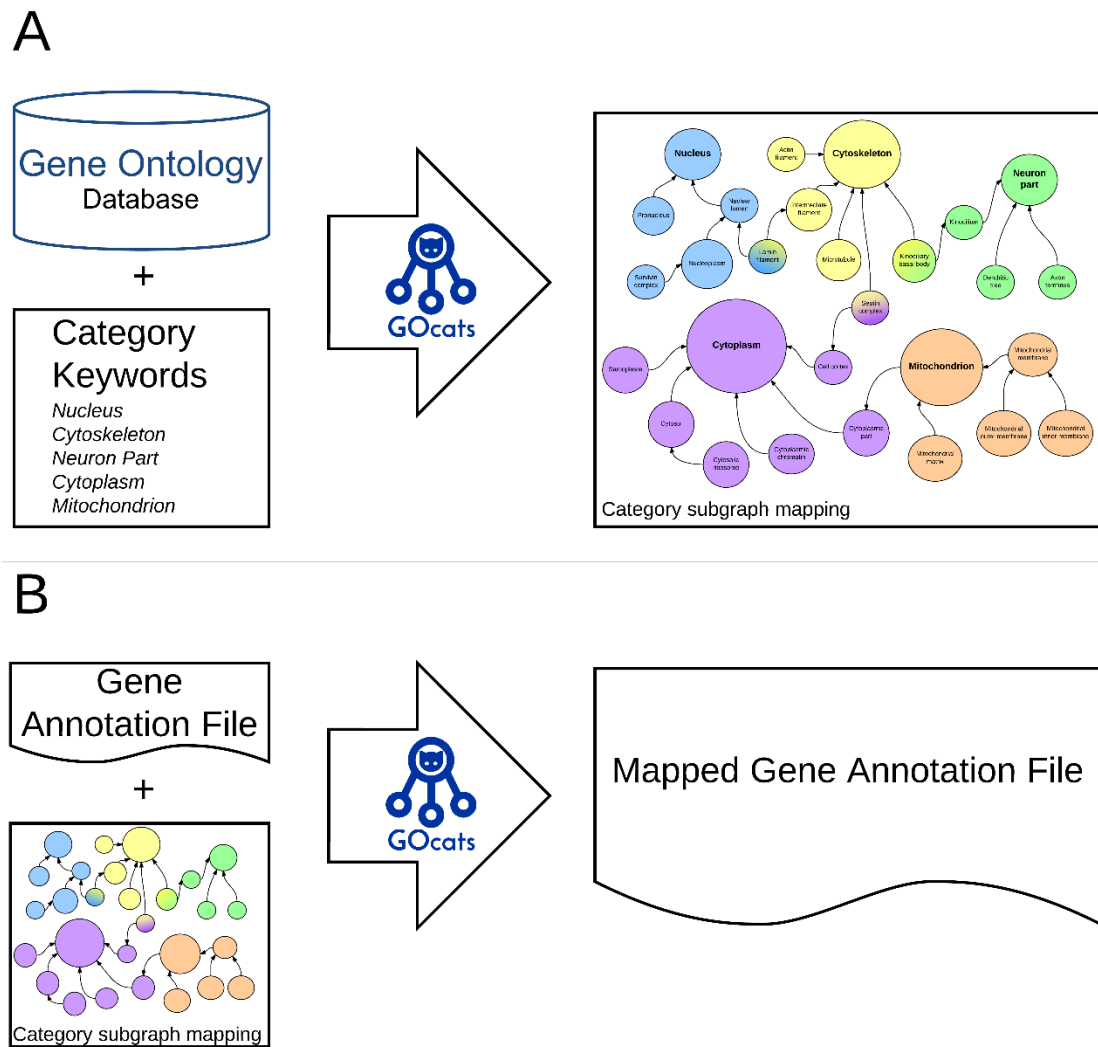


Figure 4.1 GOCats Data Flow Diagram for Creating Categories of GO

A) GOCats enables the user to extract subgraphs of GO representing concepts as defined by keywords, each with a root (category-defining) node. B) Subgraphs extracted by GOCats are used to create a mapping from all sub-nodes in a set of subgraphs to their category-defining root node(s). This allows the user to map gene annotations in GAFs to any number of customized categories.

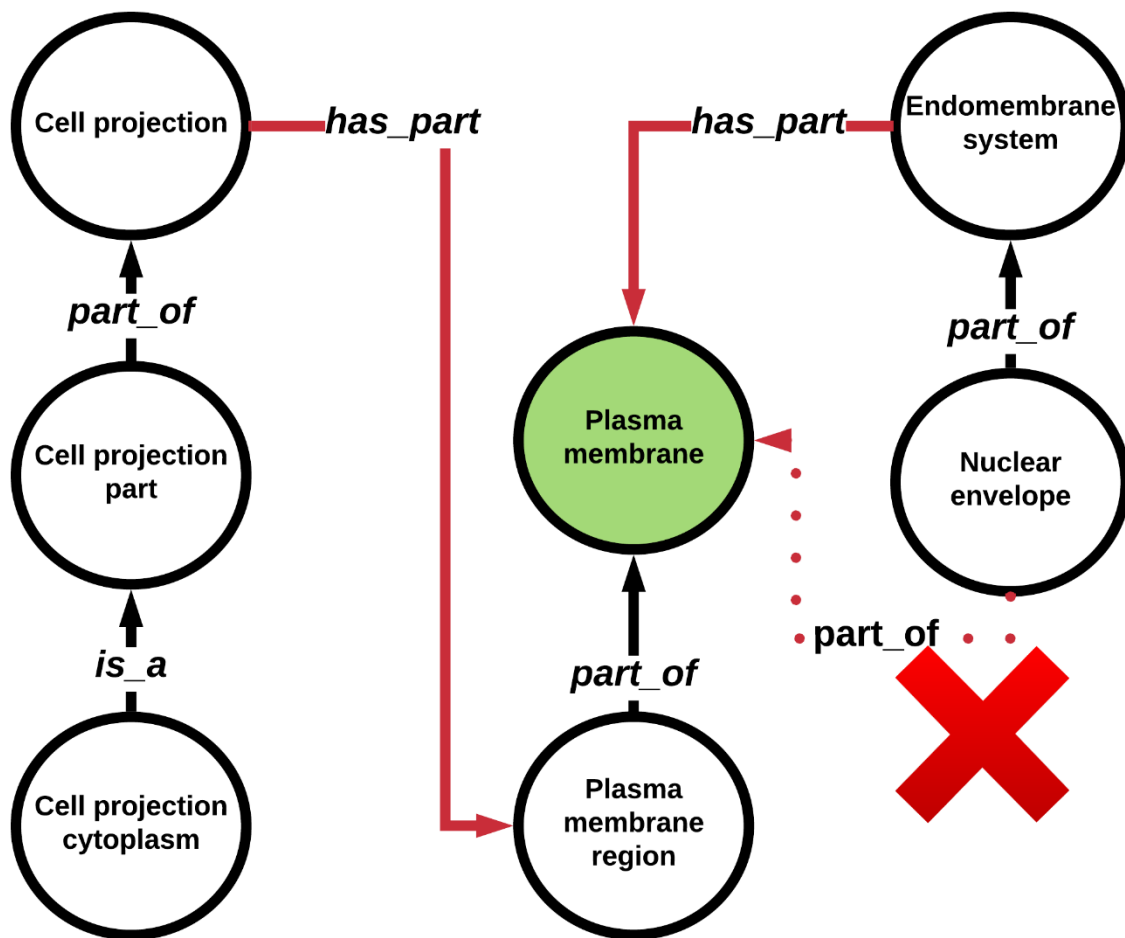


Figure 4.2 The *has\_part* Relation Creates Incongruent Paths with Respect to Semantic Scoping.

Some tools may create questionable GO term mappings, i.e. “nuclear envelope” to “plasma membrane,” since the *has\_part* relation edges point in from super-concepts to sub-concepts. GOCats avoids this by re-interpreting the *has\_part* edges into *part\_of\_some* edges.

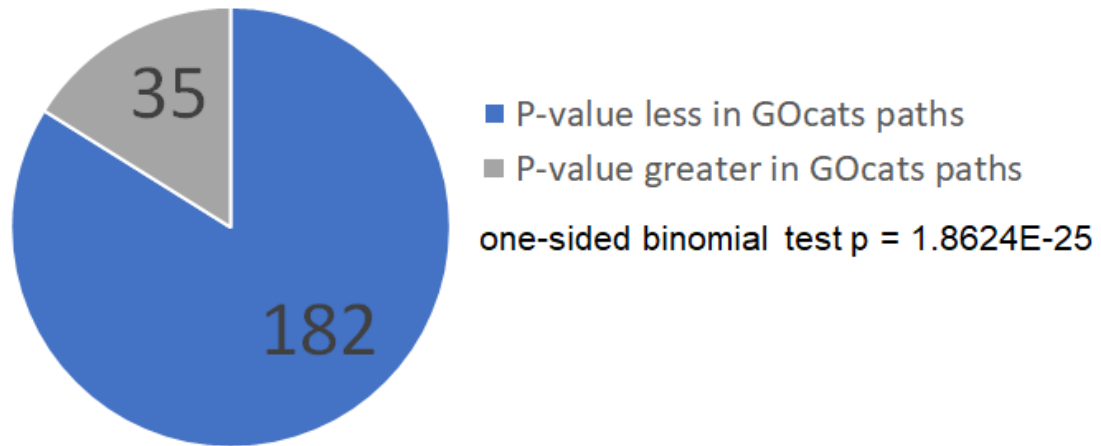


Figure 4.3 Comparison of Adjusted p-values for Significantly-enriched Annotations Using GOcats Paths vs Excluding *has\_part* Edges

Most significantly-enriched GO terms had an improved p-value when GOcats re-evaluated *has\_part* edges for the enrichment of the breast cancer data set in this investigation.

## CHAPTER 5. ANNOTATION ENRICHMENT ANALYSIS APPLICATIONS

### 5.1 Identifying Enriched Annotations and Putative Gene Targets among Differentially-expressed Genes during the Fetal Developmental Progression of Equine Tissue

#### 5.1.1 Background and Experimental Design

As described in Chapter sections 4.5.5 and 4.5.6, we used GOcats along with CategoryCompare2 to perform annotation enrichment for DESeq2 RNAseq datasets derived from developing equine cartilaginous tissues (interzone and anlagen) across two gestational time points and their neonatal derivatives, articular cartilage and epiphyseal cartilage, respectively, in collaboration with Dr. James MacLeod and Dr. Rashmi Dubey. The immediate goal was to identify enriched annotations between each time point along the developmental process to determine what molecular functions, biological processes, and cellular locations are characteristic of anlagen and articular cartilage development. These results would then be leveraged to identify key regulatory drivers of development and differentiation in each tissue type.

As previously described, enrichment was performed as pairwise analyses between each time point for each tissue type: fetal interzone tissue at 45 days of gestation (iz\_45); fetal anlagen tissue at 45 days (anl\_45); fetal interzone tissue at 60 days of gestation (iz\_60); anlagen fetal tissue at 60 days (anl\_60); neonatal articular cartilage (ac\_neo); and neonatal epiphyseal cartilage (epi\_neo). For each pairwise time point comparison from the DESeq2 analyses (iz/anl\_45-iz/anl\_60, iz/anl\_60-ac/Epi\_neo, or iz/anl\_45-ac/Epi\_neo), we selected positively- or negatively-changing genes by filtering to those changing genes

which had an adjusted p-value  $< 0.01$ . Based on the sign of each gene's fold expression from the dataset, we classified these genes into categories for categoryCompare2 as “positive”, “negative”, or “all” (either positively or negatively changing in expression). Enrichment was performed on each of these three categories for each three pairwise time point comparisons for each of the two tissue types, yielding 18 total enrichment analyses. Annotations were obtained from transcript IDs by mapping them to gene annotations available from AgBase (78). Enrichment was performed while utilizing the full ontological ancestor paths for each annotation using GOcats' path tracing algorithms. Details describing the Snakemake workflows that streamline the combined use of GOcats and CategoryCompare2 for this time-series analysis can be found in chapter 2.2.

In addition to individual pairwise time point analyses, we also defined special gene sets relating to the scope of the whole time series. These were defined as i) early - those genes that significantly increased or decreased in fold-change during the iz/anl\_45-iz/anl\_60 time point comparison, but did not significantly change in the iz/anl\_60-ac/epi\_neo time point comparison; ii) late - those genes that did not have a significant fold-change in the iz/anl\_45-iz/anl\_60 time point comparison, but did significantly change in the iz/anl\_60-ac/epi\_neo time point comparison; iii) transient - those genes that significantly change during the iz/anl\_45-iz/anl\_60 time point comparison, but then significantly change in the opposite direction during the iz/anl\_60-ac/epi\_neo time point comparison; and iv) consistent - those genes that experience fold change in expression consistently throughout the time series. We also divided each of these whole time series gene sets into positive and negative sets corresponding to the sign of the fold-change. In

the case of transient, the directionality corresponds to the fold change in the first, iz/anl\_45-iz/anl\_60 time point comparison.

### 5.1.2 Results

The resulting enrichment tables from the 42 total enrichment analyses performed among all pairwise and whole time series comparisons can be found in our FigShare repository (83). To summarize the results derived from the anlagen tissue analyses, we observed a high enrichment of generic terms such as “system development,” “extracellular region,” and “anatomical structure development” among the “consistent” genes, which was expected considering the developmental stage at which the samples were collected and the fact that this group represented transcript expression that did not change significantly across the time series. We also observed relatively high enrichment of neuronally-relevant terms such as “nervous system development” and “neuron differentiation,” This is likely due to the high number of housekeeping genes and general, developmentally-related genes that appeared in this gene set. Interestingly, we observed high enrichment of immune system response processes that were specific to positively-expressed transcripts in the “late” development stage. These included “adaptive immune response,” “Immune effector process,” and “inflammatory response.” However, it is not clear whether these processes are directly linked to the tissue differentiation process or are simply enriched due to the developmental time point at which tissue samples were collected. In the early anlagen time point, we observed a negative expression of transcripts related to cellular components such as “contractile fiber,” “myofibril,” and “sarcomere”, suggesting an early (before 60 days gestation) down-regulation of genes affecting these cellular structures. These genes

included CAPZB, ACTA1, TMOD1, TMOD3, TMOD2, GLRX3, MYL4, SLC4A1, TNNC1, TNNC2, MYL9, TNNT1, MYH7, TPM1, CAPN3, NEXN, PPP1R12A, TTN, CFL2, SYNPO2L, XIRP2, FHOD3, MYO18B, ANK1, KLHL41, KLHL40, MYBPC1, SMPX, PPP3CB, and SCN3B. Transient genes represent those whose expression increased and then decreased throughout the entire time series. In anlagen tissue, this category was marked by cardiovascular development related terms like “angiogenesis,” “vasculature development,” and “blood vessel development.”

In the results from the interzone tissue analyses, we again observed a high enrichment of generic terms such as “response to hormone” and “cartilage development”, as well as some more specific processes related to common cellular events such as “RNA polymerase II transcription factor activity sequence-specific DNA-binding” and “transcription factor activity RNA polymerase II proximal promoter sequence-specific DNA binding” among the positively-expressed, consistent gene set. Alternatively, in the negatively-expressed consistent gene set, we observed a down-regulation of genes related to nervous system development and neuronal differentiation. Among the negatively-expressed, early gene set, we observed enrichment of “translation,” as well as several protein-localization concepts, especially targeting the membrane including “cotranslational protein targeting to membrane,” “SRP-dependent cotranslational protein targeting to membrane,” and “protein localization to endoplasmic reticulum.” Meanwhile, we did not observe any enrichment of terms among the positively expressed, early gene set. This suggests that down-regulation of genes related to protein localization may be critical for the early phases of interzone tissue differentiation. In the late interzone gene sets, we did not observe any significant or specific enriched terms; we only observed generic,

housekeeping terms related to DNA replication in the negatively expressed gene set, indicating the obvious conclusion that cell division slows at the end of the development cycle. As expected, the transient gene set enrichment results for interzone tissue was also laden with generic terms such as “organelle,” “cell,” and “intracellular part” for the gene set that initially decreased in expression, and similarly generic terms related to the cell cycle for the gene set that initially increased in expression.

As a demonstration of our ability to organize enrichment results, we generated a table of enrichment results which displays the immediate ontological parent and child terms as nested lists above and below each enriched term, respectively with their associated enrichment p-values where appropriate (Table 5.1). For better visualization, we displayed p-values as the negative decadic logarithm of the adjusted p-value (target FDR=0.01) and color coded them based on their relative enrichment in the results. This example was produced using enrichment results from the pairwise enrichment analysis of the ANL\_45 and ANL\_60 timepoints for both positively and negatively expressed transcripts.

## 5.2 Determining Features Unique to Kentucky Lung Adenocarcinoma Mutational Profiles.

### 5.2.1 Background and Experimental Design

Lung cancer has the highest morbidity of any cancer worldwide (84). In the US, the state of Kentucky ranks highest in lung cancer incidence, with an age-adjusted incidence per 100,000 of 96.8, compared to the nationwide average of 63.0 (85). To test the hypothesis that Kentucky lung cancer genomic profiles are in some way unique from the general population, colleagues within the University of Kentucky conducted the first ever

genomic characterization of lung cancers from the Appalachian region of Kentucky, and compared somatic mutational data from whole genome sequencing results to those obtained from national cohorts (86).

To briefly summarize, their study focused on squamous cell carcinoma. Tumor and non-tumor DNA samples were taken from 51 patients from the Appalachian region which were subject to whole-genome sequencing. Non-silent mutations were analyzed for their significance based on mutational frequency using MutSigCV (version 1.4) (27). These mutational frequencies were compared with mutational frequencies taken from whole - genome sequences available in The Cancer Genome Atlas (TCGA) (59)—a cohort of 178 lung squamous cell samples from patients across the US. To detect genes with mutational frequencies significantly higher in the Kentucky cohort versus the TCGA, our collaborators used a Fisher's exact test, along with a Benjamini-Hochberg procedure to calculate false discovery rates (86).

Following the publication of this study, we collaborated with two of its authors, Mr. Jinpeng Liu and Dr. Chi Wang, to analyze results from a new cohort of Kentucky lung cancer patients, now termed Kentucky Lung Cancer Genomes (KLCG). This dataset was composed of comparisons in mutational frequencies between KLCG patients and patient data from TCGA and was performed using the same methodology as the previous study. However, these results were produced from lung adenocarcinoma samples, rather than the squamous cells analyzed previously. We were interested in performing annotation enrichment analysis across the two cohorts to identify which biological concepts and processes might be unique among Kentucky adenocarcinomas.

The dataset was provided as a CSV file with one gene per row. Columns displaying the mutational frequency determined by MutSigCV for each cohort, KLCG and TCGA were also present, along with the p-value determined by the previously-mentioned Fisher's exact test for each gene. We compiled foreground genes for the KLCG cohort by selecting genes that had a higher mutational frequency in the KLCG dataset than the TCGA dataset and that also had a p-value from the Fisher's exact test lower than 0.01. Foreground genes were enriched against the universe, which was comprised of the whole set of genes in the dataset, using the enrichment methods described in Chapter 2.2.

In addition to enrichment tables, we also compiled protein-protein interaction networks by querying genes that were annotated to the terms that were highly enriched. This was done using scripts which accessed the Search Tool for the Retrieval of Interacting Genes (STRING) using STRING's REST application programming interface (see Chapter 2.3).

### 5.2.2 Results

In order to display enriched annotations cleanly within the context of potentially co-mutated or functionally-related gene sets, we grouped significantly-enriched annotations with mutually-exclusive sets to which they were annotated in the dataset. Within each group of gene supersets, the annotations are listed in order of decreasing adjusted p-value (target FDR = 0.01). These results are displayed in Table 5.2, where gene supersets—sets of genes that share mutually-exclusive, significantly enriched GO annotations (adjusted  $p < 0.01$ )—are listed in the merged green cells, and their associated GO term enrichments are listed below.

Curiously, the largest superset of genes was associated with enriched annotations relating to cardiovascular neuromuscular signaling, comprising 14 genes and 20 significantly-enriched annotations. The group with the second largest number of associated annotations was comprised of the genes *EIF2AK3* and *EIF2AK4*, which are associated with seven terms related cellular stress response. Another promising hit are the genes associated with adrenergic receptor binding: *APLP1*, *ARRB1*, *NEDD4*, and *GNAS*, as evidence continues to mount regarding adrenergic receptors' role in lung cancer (87).

Concepts seemingly unrelated to lung cancer such as “regulation of oogenesis” and perhaps even the largest category including cardiac neuromuscular signaling may be attributed to idiosyncrasies among the sample population in the KLCG cohort. In other words, considering that whole-genome sequencing was used in the study, it is possible that somatic mutational frequencies unrelated to mutations driving adenocarcinoma were detected among the local population sampled in the KLCG cohort.

Using the gene supersets identified when grouping mutually-exclusive sets of enriched GO terms, we queried the STRING (55) database to find known and predicted interaction networks involving the identified genes (see Chapter 2.3). As expected with genes grouped by functional annotations, highly-connected networks of known and predicted protein interactions were observed after a single iteration of additional nodes were added in STRING (Figure 5.1 and 5.2). Of the cardiovascular-related gene superset, we found that *CACNAIS*, *CACNAIG*, and *CACNAII* formed the center of the largest connected portion of the network (Figure 5.1). STRING's functional enrichment analysis identified 11 of these 12 nodes as enriched for “ion gated channel activity,” complementing our enrichment results. Meanwhile, in the protein-protein interaction network produced for

adrenergic receptor binding proteins, a node that STRING added to the network served as the hub, *ARDB2*: beta-2 adrenergic receptor (Figure 5.2). The pink edge connecting this protein with mutated genes in this dataset, *NEDD4*—an E3 ubiquitin ligase, *ARRB*—a regulator of agonist-mediated G protein coupled receptor signaling, and *GNAS*—the stimulatory alpha subunit of G protein indicates experimentally determined, known interactions. Additional protein-protein interaction networks for the remaining gene supersets are available in tabular format on our FigShare repository (88).

While these results serve to guide further investigation into the factors driving the increased incidence of lung cancer in Kentucky, we wish to use this application to highlight the versatility that our tool affords the scientific community. In this demonstration, GOcats is integrated seamlessly within data analysis pipelines with increased complexity. Here, using simple scripts within the Snakemake workflow management system, GOcats' augmented ontological path traversal algorithms can be integrated with annotation enrichment software, and enriched annotations can be leveraged to identify potential interacting proteins. These identified proteins can be queried with other tools like STRING, utilizing their available REST APIs.

As information is compiled digitally within online repositories with increased frequency and volume, we envision that the ability to synchronize tools for distilling and leveraging this information toward solving scientific questions will become increasingly more valuable. This is why we are not only interested in utilizing a fuller extent of the knowledge available in ontologies like GO but are also driven to create open-sourced and well-documented command-line-implemented software tools that can be integrated into complex data analysis pipelines. Such an effort, we hope, will benefit the scientific

community by providing increased freedom to custom-tailor scalable and reproducible analyses in the age of “big data.”

Table 5.1 Enrichment Results of ANL\_45/ANL\_60 Pairwise Time-series Comparison for Positively and Negatively Expressed Transcripts, Nested to Show Enrichment of Parent and Child GO Terms

Parent GO Term	Enriched GO term	Child GO Term	Term Name	Enrichment Score -1 * log10(padjust)
GO:0005575			cellular_component	3.045586915
GO:0005576			extracellular region	15.88165181
	*GO:0044421		extracellular region part	15.88165181
		GO:0031012	extracellular matrix	5.419450707
		GO:0005615	extracellular space	14.62882797
		GO:0043230	extracellular organelle	11.5465223
GO:0005575			cellular_component	3.045586915
	*GO:0005576		extracellular region	15.88165181
		GO:0044421	extracellular region part	15.88165181
GO:0048856			anatomical structure development	12.35823978
GO:0032502			developmental process	12.91040648
	*GO:0009653		anatomical structure morphogenesis	14.72401342
		GO:0048598	embryonic morphogenesis	2.834108421
		GO:0035239	tube morphogenesis	7.192389905
		GO:0048646	anatomical structure formation involved in morphogenesis	4.981839946
		GO:0022603	regulation of anatomical structure morphogenesis	8.465784815
		GO:0032989	cellular component morphogenesis	4.699408113
		GO:0009887	animal organ morphogenesis	7.385570738
		GO:0048729	tissue morphogenesis	4.539976396
GO:0044421			extracellular region part	15.88165181
	*GO:0005615		extracellular space	14.62882797
		GO:0070062	extracellular exosome	11.4548025
GO:0048856			anatomical structure development	12.35823978
GO:0048731			system development	12.91040648
	*GO:0048513		animal organ development	13.36794458
		GO:0060485	mesenchyme development	4.891109643
		GO:0007423	sensory organ development	2.495327041
		GO:0009887	animal organ morphogenesis	7.385570738

		GO:0030323	respiratory tube development	2.395968026
		GO:0001822	kidney development	5.432238868
		GO:0048568	embryonic organ development	2.59086998
		GO:0051216	cartilage development	4.256882426
		GO:0030324	lung development	2.495327041
		GO:0007420	brain development	2.3922436
		GO:0060348	bone development	5.245113228
		GO:0007507	heart development	3.872241758
GO:0008150			biological process	2.250719589
	*GO:0032502		developmental process	12.91040648
		GO:0048869	cellular developmental process	7.502195616
		GO:0051093	negative regulation of developmental process	4.59473031
		GO:0050793	regulation of developmental process	6.462111444
		GO:0048646	anatomical structure formation involved in morphogenesis	4.981839946
		GO:0048856	anatomical structure development	12.35823978
		GO:0051094	positive regulation of developmental process	4.539267389
		GO:0009653	anatomical structure morphogenesis	14.72401342
GO:0048856			anatomical structure development	12.35823978
GO:0007275			multicellular organism development	12.59459121
	*GO:0048731		system development	12.91040648
		GO:0060541	respiratory system development	2.099836782
		GO:0072358	cardiovascular system development	8.105726276
		GO:0001944	vasculature development	8.044546974
		GO:0001655	urogenital system development	5.515275258
		GO:0001501	skeletal system development	6.736505449
		GO:0007417	central nervous system development	3.429519496
		GO:0048513	animal organ development	13.36794458
		GO:0072359	circulatory system development	9.526431307
		GO:0007399	nervous system development	4.564314103
		GO:0072001	renal system development	5.554539988
GO:0048856			anatomical structure development	12.35823978

GO:0032501			multicellular organismal process	7.114525733
	*GO:0007275		multicellular organism development	12.59459121
		GO:0046661	male sex differentiation	2.119420753
		GO:2000026	regulation of multicellular organismal development	6.293635737
		GO:0048731	system development	12.91040648
		GO:0009790	embryo development	2.882079966
		GO:0035295	tube development	8.105726276
GO:0032502			developmental process	12.91040648
	*GO:0048856		anatomical structure development	12.35823978
		GO:0030900	forebrain development	2.564714972
		GO:0001568	blood vessel development	7.221338988
		GO:0048513	animal organ development	13.36794458
		GO:0048468	cell development	5.654074116
		GO:0048839	inner ear development	2.54300648
		GO:0007275	multicellular organism development	12.59459121
		GO:0009888	tissue development	11.4548025
		GO:0032835	glomerulus development	2.757171443
		GO:0035295	tube development	8.105726276
		GO:0061061	muscle structure development	4.681644792
		GO:0048731	system development	12.91040648
		GO:0009653	anatomical structure morphogenesis	14.72401342
		GO:0060322	head development	3.239951832

Table 5.2 Enriched annotations among genes with higher mutational frequency in the KLCG cohort versus the TCGA cohort

Gene superset	GO Term	Description	Ontology Namesapce	Enrichment adjusted p-value	Odds ratio	Associated genes from this dataset
{ 'SCN5A', 'CACNA1S', 'CACNA1G', 'KCNMB4', 'SHISA9', 'GABRA2', 'KCND2', 'ABCA2', 'CUBN', 'CACNA1I', 'LRRC38', 'HCN2', 'ATP1A3', 'CACNG4' }						
	GO:0019228	neuronal action potential	biological process	0.00023226	10.5981219	CACNA1I;KCNMB4;KCND2;SCN5A;CACNA1G
	GO:0008332	low voltage-gated calcium channel activity	molecular function	0.00133456	92.5578635	CACNA1I;CACNA1G
	GO:0090676	calcium ion transmembrane transport via low voltage-gated calcium channel	biological process	0.00263156	46.2759644	CACNA1I;CACNA1G
	GO:0086010	membrane depolarization during action potential	biological process	0.00313866	7.43737313	CACNA1I;SCN5A;HCN2;CACNA1G
	GO:0051899	membrane depolarization	biological process	0.00342211	5.41498399	CACNA1I;SCN5A;CACNG4;HCN2;CACNA1G
	GO:0140200	adenylate cyclase-activating adrenergic receptor signaling pathway involved in regulation of heart rate	biological process	0.00356167	7.15086108	CACNA1I;SCN5A;HCN2;CACNA1G
	GO:0086045	membrane depolarization during av node cell action potential	biological process	0.0043243	30.8486647	SCN5A;CACNA1G
	GO:0086046	membrane depolarization during sa node cell action potential	biological process	0.0043243	30.8486647	SCN5A;CACNA1G
	GO:0086012	membrane depolarization during cardiac muscle cell action potential	biological process	0.00434943	10.7032967	SCN5A;HCN2;CACNA1G
	GO:0086023	adenylate cyclase-activating adrenergic receptor signaling pathway involved in heart process	biological process	0.00452046	6.63923241	CACNA1I;SCN5A;HCN2;CACNA1G
	GO:1902495	transmembrane transporter complex	cellular component	0.00481981	2.32266522	CACNA1I;ATP1A3;LRRC38;KCNMB4;SHISA9;KCND2;ABCA2;SCN5A;CACNG4;CUBN;CACNA1S;GABRA2;HCN2;CACNA1G

	GO:1990351	transporter complex	cellular component	0.00556553	2.28173543	CACNA1I;ATP1A3;LRR C38;KCNMB4;SHISA9; KCND2;ABCA2;SCN5A; CACNG4;CUBN;CACN A1S;GABRA2;HCN2;CA CNA1G
	GO:0003062	regulation of heart rate by chemical signal	biological process	0.0056392	6.1958209	CACNA1I;SCN5A;HCN2 ;CACNA1G
	GO:0086103	g-protein coupled receptor signaling pathway involved in heart process	biological process	0.00626199	5.99557053	CACNA1I;SCN5A;HCN2 ;CACNA1G
	GO:0086027	av node cell to bundle of his cell signaling	biological process	0.00882831	18.5068249	SCN5A;CACNA1G
	GO:0060371	regulation of atrial cardiac muscle cell membrane depolarization	biological process	0.00882831	18.5068249	SCN5A;CACNA1G
	GO:0086016	av node cell action potential	biological process	0.00882831	18.5068249	SCN5A;CACNA1G
	GO:0098874	spike train	biological process	0.00949279	3.53928612	CACNA1I;KCNMB4;KC ND2;SCN5A;HCN2;CAC NA1G
	GO:0001508	action potential	biological process	0.00949279	3.53928612	CACNA1I;KCNMB4;KC ND2;SCN5A;HCN2;CAC NA1G
	GO:0034703	cation channel complex	cellular component	0.00999474	2.49126948	CACNA1I;LRRC38;KCN MB4;SHISA9;KCND2;S CN5A;CACNG4;CACNA 1S;HCN2;CACNA1G
{EIF2AK3', 'EIF2AK4'}						
	GO:0036491	regulation of translation initiation in response to endoplasmic reticulum stress	biological process	0.00133456	92.5578635	EIF2AK3;EIF2AK4
	GO:0032057	negative regulation of translational initiation in response to stress	biological process	0.00263156	46.2759644	EIF2AK3;EIF2AK4
	GO:0004694	eukaryotic translation initiation factor 2alpha kinase activity	molecular function	0.00263156	46.2759644	EIF2AK3;EIF2AK4
	GO:0010998	regulation of translational initiation by eif2 alpha phosphorylation	biological process	0.00263156	46.2759644	EIF2AK3;EIF2AK4

	GO:0036490	regulation of translation in response to endoplasmic reticulum stress	biological process	0.00639548	23.1350148	EIF2AK3;EIF2AK4
	GO:0032055	negative regulation of translation in response to stress	biological process	0.00639548	23.1350148	EIF2AK3;EIF2AK4
	GO:0070417	cellular response to cold	biological process	0.00882831	18.5068249	EIF2AK3;EIF2AK4
{ 'UNC13A', 'RELN', 'PLA2G6', 'NTRK1', 'CUBN', 'CACNG4', 'ADORA1', 'GRM5' }						
	GO:1900451	positive regulation of glutamate receptor signaling pathway	biological process	0.00232885	13.9169643	UNC13A;CACNG4;RELN
	GO:0051968	positive regulation of synaptic transmission; glutamatergic	biological process	0.00402178	6.88557214	CACNG4;CUBN;NTRK1;RELN
	GO:0051966	regulation of synaptic transmission; glutamatergic	biological process	6.78E-05	6.58928287	UNC13A;CACNG4;CUBN;ADORA1;GRM5;PLA2G6;NTRK1;RELN
{ 'IL4R', 'CACNA1G', 'PLA2G6', 'CDK5R2', 'CACNA1I', 'GATA2' }						
	GO:1903307	positive regulation of regulated secretory pathway	biological process	0.00232566	5.97190235	CACNA1I;CDK5R2;GATA2;CACNA1G;IL4R
	GO:0045921	positive regulation of exocytosis	biological process	0.00456258	4.17641522	CACNA1I;CDK5R2;PLA2G6;GATA2;CACNA1G;IL4R
	GO:0045956	positive regulation of calcium ion-dependent exocytosis	biological process	0.00955052	7.72767857	CACNA1I;CDK5R2;CACNA1G
{ 'MUC12', 'MUC5B', 'MUC19', 'MUC20', 'MUC3A' }						
	GO:0016266	o-glycan processing	biological process	0.00616309	4.65479042	MUC5B;MUC12;MUC3A;MUC20;MUC19
	GO:0002223	stimulatory c-type lectin receptor signaling pathway	biological process	0.00950897	4.15445894	MUC5B;MUC12;MUC3A;MUC20;MUC19
{ 'APLP1', 'ARRB1', 'NEDD4', 'GNAS' }						
	GO:0031690	adrenergic receptor binding	molecular function	0.00029903	15.5074627	NEDD4;ARRB1;APLP1;GNAS
	GO:0031698	beta-2 adrenergic receptor binding	molecular function	0.00263156	46.2759644	NEDD4;GNAS
{ 'DYNC2H1', 'TRIM58', 'DNAH10', 'DNAH6' }						

	GO:0045505	dynein intermediate chain binding	molecular function	0.00239791	8.08513952	DNAH10;TRIM58;DYNC2H1;DNAH6
	GO:0008569	atp-dependent microtubule motor activity; minus-end-directed	molecular function	0.00614164	9.275	DNAH10;DYNC2H1;DNAH6
{ 'IGF1', 'NTRK1', 'ATP1A3', 'GRM5' }						
	GO:1904646	cellular response to amyloid-beta	biological process	0.00313866	7.43737313	ATP1A3;IGF1;GRM5;NTRK1
	GO:1904645	response to amyloid-beta	biological process	0.00452046	6.63923241	ATP1A3;IGF1;GRM5;NTRK1
{ 'GATA2', 'JAGN1', 'FASN' }						
	GO:0030223	neutrophil differentiation	biological process	0.00133456	92.5578635	JAGN1;FASN
	GO:0030851	granulocyte differentiation	biological process	0.00358998	11.5959821	JAGN1;GATA2;FASN
{ 'IGF1', 'IRS2', 'DYRK2' }						
	GO:0045725	positive regulation of glycogen biosynthetic process	biological process	0.00434943	10.7032967	IGF1;IRS2;DYRK2
	GO:0070875	positive regulation of glycogen metabolic process	biological process	0.00519906	9.93813776	IGF1;IRS2;DYRK2
{ 'IGF1', 'WEE2' }						
	GO:1905880	negative regulation of oogenesis	biological process	0.00263156	46.2759644	IGF1;WEE2
	GO:0060283	negative regulation of oocyte development	biological process	0.00263156	46.2759644	IGF1;WEE2
{ 'XDH', 'SEMA6A' }						
	GO:1900747	negative regulation of vascular endothelial growth factor signaling pathway	biological process	0.00639548	23.1350148	XDH;SEMA6A
	GO:1902548	negative regulation of cellular response to vascular endothelial growth factor stimulus	biological process	0.00882831	18.5068249	XDH;SEMA6A
{ 'UNC13A', 'RELN', 'NTRK1', 'CUBN', 'CA7', 'EIF2AK4', 'ADORA1', 'CACNG4' }						

	GO:0050806	positive regulation of synaptic transmission	biological process	0.00864377	2.92088369	UNC13A;CACNG4;CUBN;ADORA1;CA7;NTRK1;EIF2AK4;RELN
{ 'GNAS', 'SALL3', 'TFAP2B', 'TBX3' }						
	GO:0035137	hindlimb morphogenesis	biological process	0.00505914	6.40988163	TFAP2B;SALL3;TBX3;GNAS
{ 'ARHGEF2', 'RELN', 'SEMA6A' }						
	GO:2001224	positive regulation of neuron migration	biological process	0.00181994	15.4642857	SEMA6A;ARHGEF2;RELN
{ 'IGF1', 'ARRB1', 'GNAS' }						
	GO:0005159	insulin-like growth factor receptor binding	molecular function	0.0029176	12.650974	IGF1;ARRB1;GNAS
{ 'GATA2', 'NKX6-2', 'SOX1' }						
	GO:0021514	ventral spinal cord interneuron differentiation	biological process	0.00519906	9.93813776	NKX6-2;SOX1;GATA2
{ 'RBMXL2', 'SNRPA', 'RBMXL3' }						
	GO:0000243	commitment complex	cellular component	0.00717967	8.69475446	RBMXL3;RBMXL2;SNRPA
{ 'SLC15A4', 'SLC25A29' }						
	GO:0089709	L-histidine transmembrane transport	biological process	0.00133456	92.5578635	SLC25A29;SLC15A4
{ 'STRA6', 'TFAP2B' }						
	GO:0097070	ductus arteriosus closure	biological process	0.0043243	30.8486647	TFAP2B;STRA6
{ 'EGR3', 'SOX13' }						
	GO:0045586	regulation of gamma-delta t cell differentiation	biological process	0.00639548	23.1350148	SOX13;EGR3
{ 'PPP1R16B', 'GATA2' }						
	GO:1903589	positive regulation of blood vessel endothelial cell proliferation involved in sprouting angiogenesis	biological process	0.00882831	18.5068249	PPP1R16B;GATA2

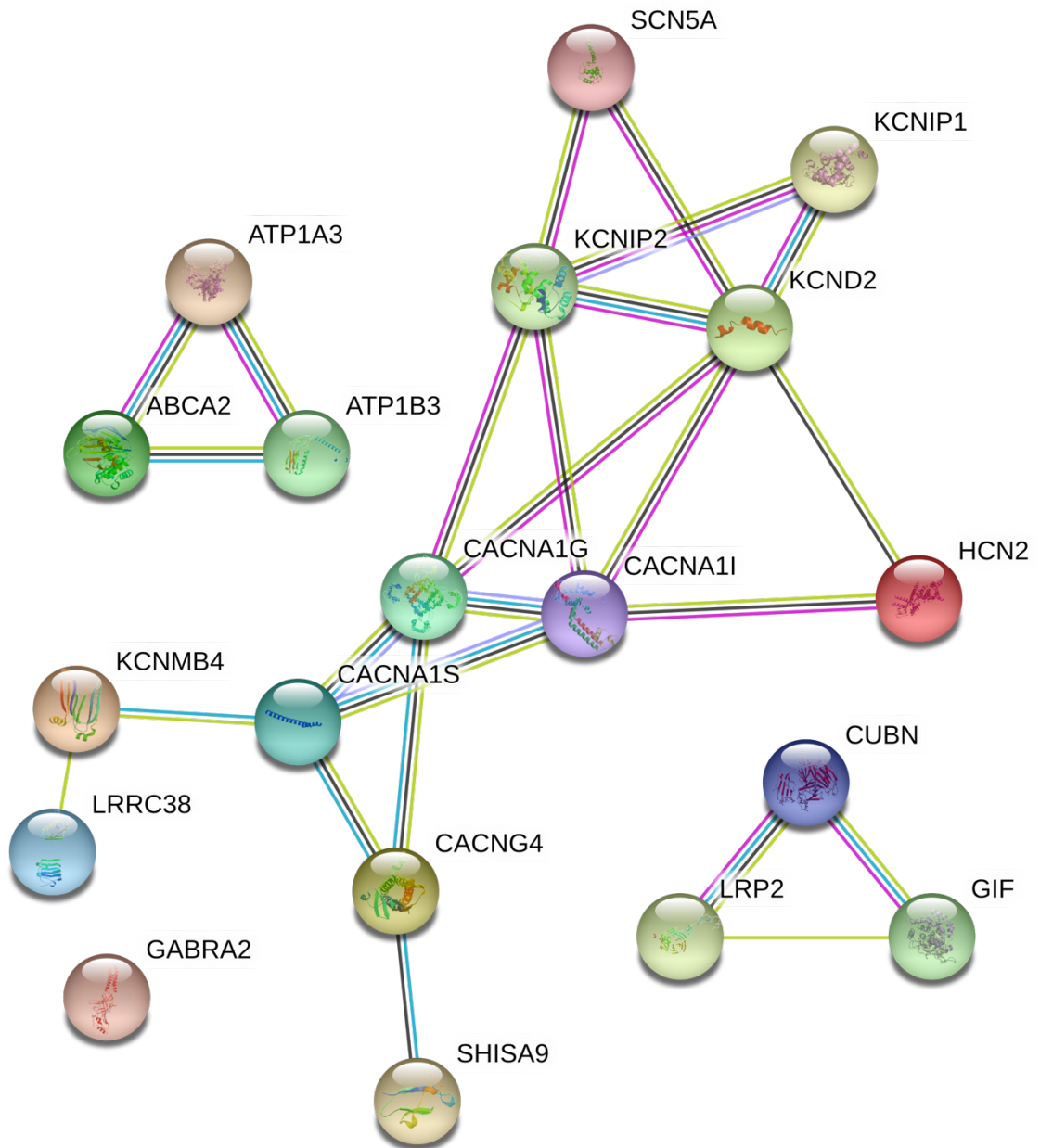


Figure 5.1 Protein-protein interaction network produced after one iteration of additional nodes in STRING from a query of *SCN5A*, *CACNA1S*, *CACNA1G*, *KCNMB4*, *SHISA9*, *GABRA2*, *KCND2*, *ABCA2*, *CUBN*, *CACNA1I*, *LRRC38*, *HCN2*, *ATP1A3*, and *CACNG4*

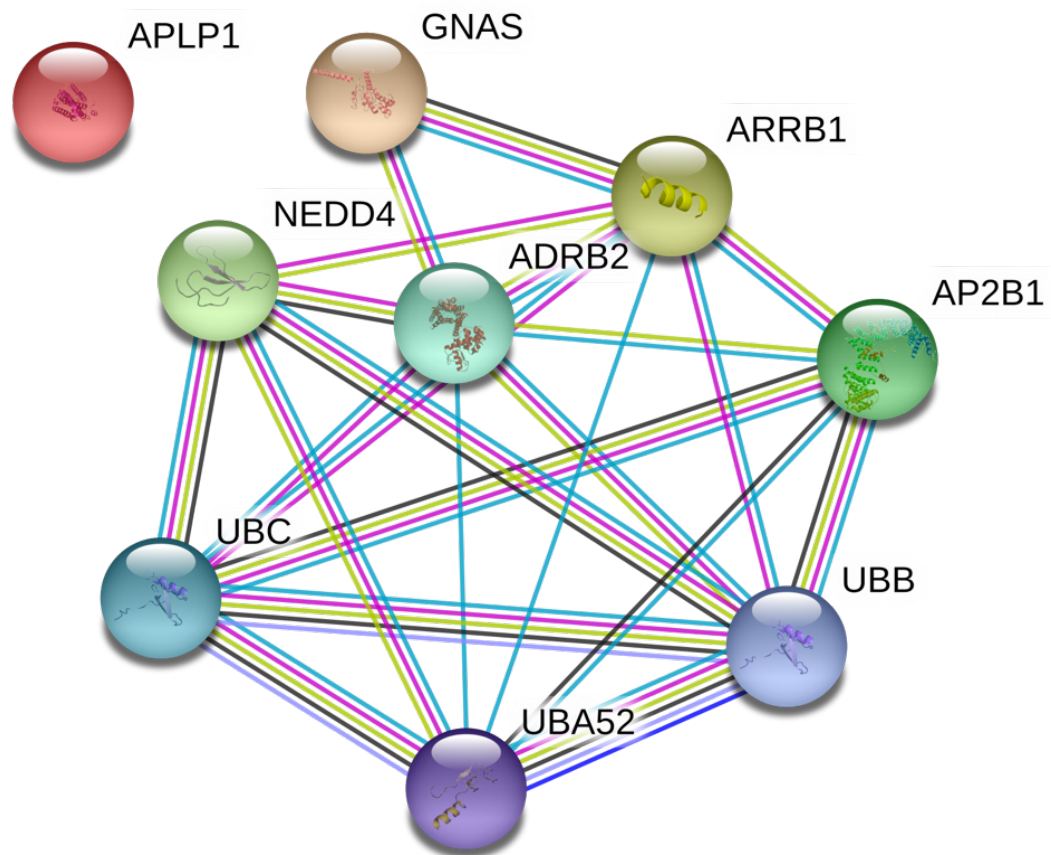


Figure 5.2 Protein-protein interaction network produced after one iteration of additional nodes in STRING from a query of *APLP1*, *ARRB1*, *NEDD4*, and *GNAS*

## CHAPTER 6. FUTURE DIRECTIONS

The work demonstrated in this dissertation represent critical first steps toward our goal of automating the enumeration of compactly-represented concepts in GO. We envision developing a tool which can automatically identify and delineate subgraphs of GO that represent distinct biologically-relevant concepts that are optimized to the granularity of available annotated data and knowledge. We plan on enabling the tool to use rank-frequency based metrics to select candidate keywords from the GO corpus to represent broad concepts. From these keywords, the tool will build subgraphs in a similar manner to GOcats' current handling of user-supplied keywords. Finally, the tool will apply metrics to test the quality of each identified subgraph before finalizing the identification of each compactly-represented concept subgraph. These metrics will include relative subgraph density and size, enrichment of semantically-similar words within each subgraph, and frequency of inner-subgraph term co-occurrence among annotations in relevant knowledgebases. We plan on allowing for the adjustment of these test metrics, enabling users to fine-tune the granularity of subgraph delineation while maintaining reproducibility. Designing a version of GOcats that allows for users to input keywords to delineate subgraphs has not only given us the opportunity to quickly provide the scientific community with a helpful GO term organizing tool, but is a necessary first step toward our long-term goal of developing more automated, unsupervised methods of ontology organization. In short, these user-defined subgraphs will be used to test the accuracy of future automatically-selected subgraphs. Furthermore, we can use these subgraphs to test subgraph quality metrics. However, additional steps must be taken to further these goals. Specifically, we must enable an automated assignment of relations, classifying them on the

basis of whether they are relevant for mereological classification and we must enable robust and unsupervised methods to automatically enumerate compactly-represented concepts present in OBO-formatted ontologies as subgraphs.

## 6.1 Developing Heuristics to Automatically Assign Semantic Scaling and Scoping

### Correspondences between Annotation Terms Connected by Relationships in GO and Other Ontologies

Following the issues identified relating to the lack of descriptions and inconsistencies involving semantic scoping among ontological relationships (see Chapter subsections 1.8, 1.9 and Chapter 4), we intend to develop methods that disambiguate scoping and scaling correspondences among ontological relationships. We hypothesize that, through combining the use of natural language processing (NLP) and the traversal of the Relations Ontology (RO) (1), we will be able to automatically classify semantic correspondences for each relationship in RO such that scaling, scoping and other relationship correspondences can be used to inform ontology term categorization in a way that enables an automatic and unbiased categorization of GO terms. These relation classes will be used to disambiguate the type of relationship encountered in any ontology which uses relationships contained in RO, so that term categorization can occur by a thorough evaluation of the semantics of given relationships, and not by making assumptions of relationship edge directionality or omitting relationships which have problematic scoping correspondences (Figure 4.2), as such omissions limit the amount of information available from ontologies (Table 4.2).

### 6.1.1 Defining relationship correspondence classes.

The following five general classes for relational correspondence will stand as a starting point for reducing ambiguity (See Table 4.1): scoping (hyponym-hypernym), scaling (meronym-holonym), spatiotemporal (process-process, process-entity, entity-entity), active (actor- subject), and equivalence. These general classes were determined based on an initial evaluation of the relationships contained within GO: *is\_a*, *part\_of*, *has\_part*, *regulates*, *positively\_regulates*, *negatively\_regulates*, *starts\_during*, *ends\_during*, *occurs\_in*, and *never\_in\_taxon*. A sixth class, other, will also be used to bin relationships that do not meet the other criteria and will serve to inform us on how to improve categories or if additional classes need to be created.

### 6.1.2 Parsing and classifying relationships in the Relations Ontology.

RO, like GO, is a graph with nodes and edges; except in RO, nodes represent semantic relationships and edges are also semantic relationships that define how two relationships are related to one another. We have already developed methods to read, parse, and create graph objects from obo-formatted ontologies like the RO. Furthermore, our current methods have already proven successful at extracting and categorizing subgraphs from such ontologies given a set of criteria. Therefore, we are in a favorable position to evaluate the RO, and extract and categorize relationships into the aforementioned groups using logical heuristics that rely on NLP.

NLP tools such as the Natural Language Tool Kit (NLTK) (89) can easily and automatically identify parts- of-speech. Auxiliary verbs (like “has,” “can,” and “is”) can

be distinguished from lexical verbs (like “regulates,” or “innervates”), simplifying, for example, the distinction of the scoping and scaling classes (with many auxiliary verbs and few or no lexical verbs) from the active classifier (prominent lexical verbs). NLP is also capable of more complex and sophisticated processing, and we anticipate utilizing advanced features to evaluate definition lines and example phrases contained for many of the term entries in RO. We do not anticipate that NLP will be a computationally expensive task, especially considering the size of RO. However, we can use the relational logic of the RO graph to decrease the amount of NLP computations; if a relationship term is determined to be active, for example, then all of its parent terms up until a branch point in the graph should be considered active without requiring redundant, individual evaluation. Then, we will test the predicted relationship directional logic derived from the NLP analysis directly against a specific ontology to validate the ontology-specific result. Using the relationships in GO, we can then validate the whole approach.

One potential problem with this approach is that certain relationship types might not be uniformly utilized from a directional perspective across all ontologies in the OBO. In other words, one ontology may utilize the same relationship type in a slightly but significantly different way from another. This may necessitate the development of methods that compare relationship directional utilization between ontologies to detect inconsistent directionality of specific relationship types within certain ontologies. A simple example of such a situation is if one ontology consistently places the directionality of the *regulates* relation from an entity to a process, whereas another consistently places the directionality of this relation from a process to an entity. Another similar situation could arise if both of these utilizations occur within a single ontology. The former situation could be rectified by

independently determining the direction of relationships for each ontology prior to correspondence classification while the latter represents a fundamental error in the ontological framework that should require the attention of the ontology’s curators for correction.

### 6.1.3 Justification

When developing GOcats, we anticipated that some relationships in GO, such as *has\_part* would prove problematic in aggregating terms into common categories due to the fact that the edge pointed from a whole entity to its part, opposing the usual semantic directionality with respect to the granularity encountered in GO. Therefore, we created a rule whereby the scoping directionality of this particular relationship edge type was inversed during our categorization methods (See Chapter 4 and Chapter 2.1.4). Indeed, when comparing GOcats’ categories to those created by M2S, we found many examples of where the *has\_part* relationship produced questionable term categorizations when M2S was used, which were not made by GOcats. For example, the terms “nuclear envelope,” “endomembrane system,” and “cell projection cytoplasm” were all erroneously rooted to the category, “plasma membrane” by M2S, while GOcats did not make these mappings. We determined from manually examination that the error is caused by the reversed directionality of the *has\_part* relationship, and that tools such as M2S follow only edge directionality when rooting terms. Figure 4.2 shows some examples of this.

We have also calculated upper boundary estimates on the number of potential false mappings that could potentially occur from the reversed directionality of such relationships, if current mapping tools continue to take edge directionality alone into

account when mapping specific terms to general terms. We did this by calculating the number of possible mappings that could occur between applicable ancestors and descendent terms about every *has\_part* relationship (See Chapter 4.5.1 and 4.5.2). Across all of GO, we found that 121,579 potential false mappings are possible considering this relationship alone. When we compared the number of potential false mappings to total possible mappings in each of the three sub-ontologies in GO, potential false mappings accounted for 42%, 13%, and 16% of all possible mappings in cellular component, molecular function, and biological process ontologies, respectively. When simply ignoring these relationships altogether, as some methods do, we calculated a 12%, 12%, and 5% loss of information available to be gathered from the cellular component, molecular function, and biological process ontologies, respectively.

#### 6.1.4 Expected outcomes

Our current work has demonstrated the importance of defining scaling, scoping, and other semantic correspondences when categorizing ontology terms into generic concepts, as current methods have been shown to incorrectly map terms as a result of these relationships. Automating the process of determining these correspondences will enable the large-scale evaluation of RO relationships necessary to alleviate the mapping errors caused by non-conventional relationships across all OBO ontologies. Defining and later refining these heuristics will ensure that mapping tools like GOcats will be able to utilize all relationships in RO without the need for constant updating when new relationships are added in the future. Finally, it will allow GOcats to be expanded into OBOcats, a tool for

the categorization of ontological terms across ontologies in the OBO Foundry (71); all of which follow the same data structure formatting guidelines.

## 6.2 Developing Algorithms That Automatically Identify Compactly-represented Concepts in GO and Other Ontologies

We hypothesize that automating the identification of compactly-represented concepts based on scoping relationships within biological ontologies will aid in their maintenance and use by: i) indicating which concepts are less organized than others within their graphical representation, ii) allowing for an evaluation of which concepts any given ontology is equipped to annotate, unbiased by the manual definition of categories, and iii) mapping fine-grained terms to general terms in an ontology without necessitating user selection of concepts or ontology terms, thus allowing for an unbiased organization of enriched ontology terms following gene-annotation enrichment. We endeavor to combine an evaluation of the lexical composition of ontologies along with their graph structure to detect compact concepts within an ontology, quantify the degree of compactness, and allow mapping of fine-grained terms to general concepts within these subgraphs.

### 6.2.1 Defining ontological concept compactness.

Like the concept categories extracted from GO using GOCats (see Chapter 3), the concepts identified here will be represented by a subgraph of the ontology in question. The compactness of a concept will be measured by taking into account the average degree of connections among nodes in the subgraph representing that concept, the average degree of connections in the entire graph, and the amount of overlap between the subgraphs. A category that is highly compact will have a significantly higher degree of connections

among its members than the graph's average and will minimize overlap with other concepts. We intend to devise a compactness score taking these parameters into consideration in order to provide a cutoff for which concepts are significantly represented in the ontology.

#### 6.2.2 Automatic enumeration of ontological concepts via lexical analysis.

Most semantic similarity metrics used to describe semantic distances between terms in an ontology depend on the concept of information content (IC), which is related to the frequency at which a word is used; it is assumed that the less frequently a word is used, the more IC it contains (34–37). This can be appreciated by considering that the word “the” has been used over 2000 times in this text while the word “correspondence” has been used approximately 38 times; it can be inferred that the latter conveys more meaning than the former in this text. These methods usually use an external corpus to determine IC. However, our goal is to evaluate the concepts represented solely within the confines of the ontology itself. Furthermore, unlike these methods, our goal involves creating a tool to bin specific terms into previously unspecified concepts, not finding distances from terms to one or a set of manually predetermined nodes within the graph.

Zipf's law describes a statistical distribution by which the rank order of each word in a corpus is inversely proportional to its frequency and can be fit linearly on a log/log scale (90). Using each ontology as a corpus, we will fit the words contained within to a Zipf distribution to arrive at an IC scoring scheme which suits our needs to acquire candidate terms that are not too specific or too general and potentially describe biologically meaningful concepts. Using these IC scores and the inherent graph structure of the

ontology, we expect to programmatically single out concept-representative terms within any ontology. Using methods like those already developed for GOcats, we will then be able to extract subgraphs of fine-grained terms contained under each concept-representative term and define a score to describe the degree of compactness of each concept.

### 6.2.3 Justification

As shown in Chapter 3, we have evidence that GO contains distinctly separable subgraphs that describe unique biological concepts. Figure 3.1-3.3. are graphical representations of these subgraphs made using Cytoscape 3 (61) by linking all subgraph nodes (grey) to their respective category nodes (blue). Except for macromolecular complex, fine-grained terms group neatly into one or multiple concept categories. This demonstrates that although the graph structure of GO and similar ontological databases may be complex, they can still be partitioned neatly and meaningfully. These categories were partitioned using sets of user-provided keywords, similar to those we propose automatically identifying via Zipf distributions. Furthermore, when we compared the categories created by GOcats with these keywords to categories made by providing M2S with explicit GO terms of the desired categories, the categories were very similar, as evidenced by high Jaccard indices (Table 3.3). Those with large discrepancies such as “plasma membrane” could be accounted for, in part, by mapping errors encountered by M2S, as outlined in Chapter 4.2.1 and quantified in Table 4.2.

Using GOcats, we enumerated every word within the name and definition fields of every node in the Gene Ontology and plotted their absolute frequency versus rank on a log/log scale using the Matplotlib Python package (91). We plotted results from each sub-

ontology in GO (cellular\_component, biological\_process, and molecular\_function) as well as for the whole of GO. As shown in Figure 6.1, each plot shows a roughly linear distribution, consistent with what would be consistent in a Zipf or power law distribution.

#### 6.2.4 Expected outcomes

We expect these new methods to enumerate and extract many distinct concept categories within an ontology. Considering our success in extracting subgraphs of user-defined concepts from GO using GOcats, we do not expect to encounter any issues with the partitioning of subgraphs once a concept is identified. Although it is likely that additional concepts will be identified using the automated, unbiased method suggested here for categorizing genes and gene products in GO, we expect that this method will perform nearly identically to the currently implemented GOcats method when comparing the same concepts. Finally, when using this method to evaluate the structural organization of ontologies, we expect that major GO revisions will coincide with significant instances of non-compact subgraphs.

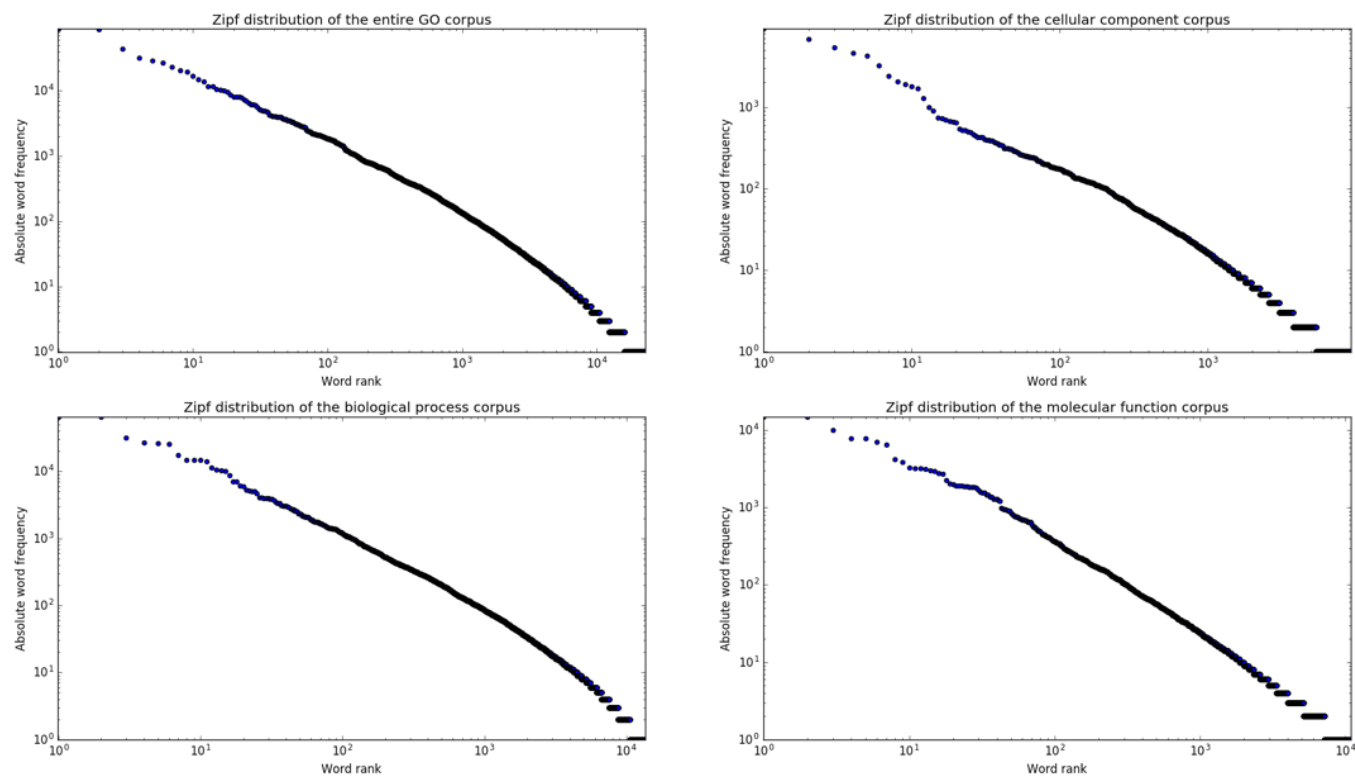


FIGURE 6.1 DISTRIBUTION OF WORD FREQUENCY VERSUS WORD RANK IN THE GENE ONTOLOGY

## REFERENCES

1. Relations Ontology [Internet]. 2016. Available from: <http://www.obofoundry.org/ontology/ro.html>
  
2. El-Sappagh S, Franda F, Ali F, Kwak KS. SNOMED CT standard ontology based on the ontology for general medical science. *BMC Med Inform Decis Mak*. 2018;18(1):1–19.
  
3. Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J. Gene Ontology: tool for the unification of biology. *Nat Genet* [Internet]. 2000;25(1):25–9. Available from: <http://www.ncbi.nlm.nih.gov/pmc/articles/pmc3037419/>
  
4. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* [Internet]. 2007;25(11):1251–5. Available from: <http://www.nature.com/doifinder/10.1038/nbt1346>
  
5. Gene Ontology consortium. Gene Ontology Consortium: going forward. *Nucleic Acids Res* [Internet]. 2015;43(D1):D1049–56. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1179>
  
6. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res* [Internet]. 2004;32(suppl 1):D267–70. Available from: [http://nar.oxfordjournals.org/content/32/suppl\\_1/D267%5Cnhttp://nar.oxfordjournals.org/content/32/suppl\\_1/D267.full.pdf%5Cnhttp://nar.oxfordjournals.org/content/32/suppl\\_1/D267.short%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/14681409](http://nar.oxfordjournals.org/content/32/suppl_1/D267%5Cnhttp://nar.oxfordjournals.org/content/32/suppl_1/D267.full.pdf%5Cnhttp://nar.oxfordjournals.org/content/32/suppl_1/D267.short%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/14681409)
  
7. Veres D V., Gyurko DM, Thaler B, Szalay KZ, Fazekas D, Korcsmaros T, et al. ComPPI: a cellular compartment-specific database for protein-protein interaction network analysis. *Nucleic Acids Res* [Internet]. 2015;43(D1):D485–93. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1007>
  
8. Huttlin EL, Ting L, Bruckner RJ, Gebreab F, Gygi MP, Szpyt J, et al. The BioPlex Network: A Systematic Exploration of the Human Interactome. *Cell* [Internet]. 2015;162(2):425–40. Available from: <http://www.sciencedirect.com/science/article/pii/S0092867415007680>

9. Papatheodorou I, Oellrich A, Smedley D. Linking gene expression to phenotypes via pathway information. *J Biomed Semantics* [Internet]. 2015;6:17. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4404592&tool=pmcentrez&rendertype=abstract>
10. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A* [Internet]. 2005;102(43):15545–50. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/16199517>
11. Na D, Son H, Gsponer J. Categorizer: a tool to categorize genes into user-defined biological groups based on semantic similarity. *BMC Genomics* [Internet]. 2014;15:1091. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4298957&tool=pmcentrez&rendertype=abstract>
12. Tang H, Klopfenstein D, Pedersen B, Flick P, Sato K, Ramirez F, et al. GOATOOLS: Tools for Gene Ontology. [cited 2016 Oct 3]; Available from: <https://zenodo.org/record/31628>
13. Chris Mungall BDGP. map2slim - maps gene associations to a “slim” ontology [Internet]. 2013. Available from: <http://search.cpan.org/~cmungall/gopherl/scripts/map2slim>
14. The UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res* [Internet]. 2015;43(D1):D204–12. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku989>
15. Cunningham F, Amode MR, Barrell D, Beal K, Billis K, Brent S, et al. Ensembl 2015. *Nucleic Acids Res* [Internet]. 2015;43(D1):D662–9. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1010>
16. OBO Principles [Internet]. [cited 2019 Mar 5]. Available from: <http://www.obofoundry.org/principles/fp-004-versioning.html>
17. Munoz-Torres M, Carbon S. Get GO! retrieving GO data using AmiGO, QuickGO, API, files, and tools. *Methods Mol Biol*. 2017;1446:149–60.

18. Spear AD, Ceusters W, Smith B. Functions in basic formal ontology. *Appl Ontol.* 2016;11(2):103–28.
19. Musen MA. The protégé project. *AI Matters.* 2015;1(4):4–12.
20. The Gene Ontology Consortium. Introduction to GO annotations [Internet]. [cited 2019 Mar 5]. Available from: <http://geneontology.org/docs/go-annotations/>
21. The Gene Ontology Consortium. Guide to GO evidence codes [Internet]. [cited 2019 Mar 5]. Available from: <http://geneontology.org/docs/guide-go-evidence-codes/>
22. Finn RD, Attwood TK, Babbitt PC, Bateman A, Bork P, Bridge AJ, et al. InterPro in 2017-beyond protein family and domain annotations. *Nucleic Acids Res.* 2017;45(D1):D190–9.
23. Mi H, Muruganujan A, Thomas PD. PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Res* [Internet]. 2013;41(D1):D377–86. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gks1118>
24. The UniProt Consortium. subcell.txt [Internet]. 2015 [cited 2015 May 27]. Available from: <http://www.uniprot.org/docs/subcell>
25. GOA Downloads [Internet]. [cited 2019 Mar 5]. Available from: <https://www.ebi.ac.uk/GOA/downloads>
26. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 2014;15(12):1–21.
27. Lawrence MS, Stojanov P, Polak P, Kryukov G V, Cibulskis K, Sivachenko A, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* [Internet]. 2013;499(7457):214–8. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23770567>  
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3919509>
28. Huang DW, Sherman BT, Lempicki RA. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 2009;37(1):1–13.

29. Flight RM, Harrison BJ, Mohammad F, Bunge MB, Moon LDF, Petruska JC, et al. Categorycompare, an analytical tool based on feature annotations. *Front Genet.* 2014;5(APR):1–13.
30. Yoav Binyamini YH. Benjamini\_Hochberg1995.Pdf. Vol. 57, *Journal of the royal statistical society.* 1995. p. 289–300.
31. Uhlen M, Fagerberg L, Hallstrom BM, Lindskog C, Oksvold P, Mardinoglu a., et al. Tissue-based map of the human proteome. *Science* (80- ) [Internet]. 2015;347(6220):1260419–1260419. Available from: <http://www.sciencemag.org/content/347/6220/1260419>
32. GO Slim and Subset Guide [Internet]. [cited 2016 Nov 22]. Available from: <http://geneontology.org/page/go-slim-and-subset-guide>
33. Binns D, Dimmer EC, Huntley RP, Barrell DG, O'Donovan C, Apweiler R. QuickGO: a web-based tool for Gene Ontology searching. *Bioinformatics* [Internet]. 2009;25(22):3045–6. Available from: <http://doi.wiley.com/10.1002/pmic.200800002>
34. Jiang JJ. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of International Conference Research on Computational Linguistics (ROCLING X).* 1997.
35. Lin D. An Information-Theoretic Definition of Similarity. In: *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning.* 1989. p. 296–304.
36. Resnik P. Semantic Similarity in a Taxonomy: An Information Based Measure and Its Application to Problems of Ambiguity in Natural Language. *J Artificial Intell Res.* 1999;11:95–130.
37. Schlicker A, Domingues FS, Rahnenführer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics* [Internet]. 2006;7:302. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-33748335463&partnerID=tZOtx3y1>
38. Abeysinghe R, Hinderer EW, Moseley HNB, Cui L. Auditing Subtype Inconsistencies among Gene Ontology Concepts. In: *The 2nd International*

Workshop on Semantics-Powered Data Analytics (SEPDA 2017) -- in conjunction with IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2017.

39. Abeysinghe R, Zheng F, Hinderer EW, Moseley HNB, Cui L. A Lexical Approach to Identifying Subtype Inconsistencies in Biomedical Terminologies. In: Quality Assurance of Biological and Biomedical Ontologies and Terminologies Workshop -- Bioinformatics and Biomedicine (BIBM), 2018 IEEE International Conference. 2018.
40. Groß A, Pruski C, Rahm E. Evolution of Biomedical Ontologies and Mappings: Overview of Recent Approaches. *Comput Struct Biotechnol J* [Internet]. 2016;14:1–8. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S2001037016300319>
41. Groß A, Dos Reis JC, Hartung M, Pruski C, Rahm E. Semi-automatic adaptation of mappings between life science ontologies. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. 2013;7970 LNBI:90–104.
42. Cesar J, Reis D, Santec CR, Tudor CRPH, Silveira M Da, Reynaud-delaître C. Mapping Adaptation Actions for the Automatic Reconciliation of Dynamic Ontologies. *Cikm*. 2013;599–608.
43. Noy N, Wallace E. Simple part-whole relations in OWL Ontologies [Internet]. W3C.org. 2005. Available from: <https://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/>
44. Gene Ontology consortium. Ontology Relations [Internet]. 2017. Available from: <http://www.geneontology.org/page/ontology-relations>
45. Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, et al. Relations in biomedical ontologies. *Genome Biol* [Internet]. 2005;6(5):R46. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1175958&tool=pmcentrez&rendertype=abstract>
46. Foundry TO. Principle: Relations [Internet]. [cited 2019 Mar 5]. Available from: <http://www.obofoundry.org/principles/fp-007-relations.html>
47. Map2Slim Documentation [Internet]. [cited 2019 Mar 5]. Available from: <https://github.com/owlcollab/owltools/wiki/Map2Slim>

48. Hinderer, Eugene W., Flight, Robert M., Dubey R, MacLeod, James N., Moseley HNB. Advances in Gene Ontology Utilization Improve Statistical Power of Annotation Enrichment. *bioRxiv*. 2018;
49. van Rossum G, Drake F. The Python Language Reference Manual. Network Theory Ltd.; 2011.
50. Hinderer EW, Moseley HNB. GOcats on GitHub [Internet]. 2017. Available from: <https://github.com/MoseleyBioinformaticsLab/Gocats>
51. Hinderer EW, Moseley HNB. GOcats on PyPi [Internet]. 2017. Available from: <https://pypi.org/project/Gocats/>
52. Gamma E, Helm R, Johnson R, Vlissides J, Booch G. Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition. Addison-Wesley Professional; 1994.
53. Hinderer EW, Moseley HNB. GOcats Documentation [Internet]. 2017 [cited 2019 Mar 5]. Available from: <https://gocats.readthedocs.io/en/latest/>
54. Köster J, Rahmann S. Snakemake-a scalable bioinformatics workflow engine. *Bioinformatics*. 2012;28(19):2520–2.
55. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, et al. STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res*. 2015;43(D1):D447–52.
56. Hinderer EW. The GOcats API Reference. Read the Docs. 2017.
57. Hinderer EW, Moseley HNB, Flight RM. Scripts and Workflows for Integrating GOcats with Annotation Enrichment Analyses [Internet]. Available from: <https://figshare.com/s/9d55b2e5932992e6a068>
58. Oliphant TE. Python for scientific computing. *Comput Sci Eng*. 2007;9(3):10–20.
59. Cancer T, Atlas G. Cancer Genome Atlas Research Network. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*. 2012;489(7417):519–25.

60. McKinney W. Data Structures for Statistical Computing in Python. Proc 9th Python Sci Conf [Internet]. 2010;1697900(Scipy):51–6. Available from: <http://conference.scipy.org/proceedings/scipy2010/mckinney.html>
61. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res [Internet]. 2003;13(11):2498–504. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=403769&tool=pmcentrez&rendertype=abstract>
62. Lex A, Gehlenborg N, Strobel H, Vuilleumot R. UpSet : Visualization of Intersecting Sets Supplementary Material. IEEE Trans Vis Comput Graph. 2014;20(12):1983–1992.
63. Hinderer EW, Moseley HNB. Scripts for the Categorization of Genes by Their Annotations Using GOcats and Other Methods [Internet]. 2017 [cited 2019 Mar 5]. Available from: <https://figshare.com/s/73b8f454516f0e7cbcc7>
64. pyUpSet [Internet]. 2016. Available from: <https://github.com/ImSoErgodic/py-upset>
65. Huber W, Gentleman R. estrogen: Microarray dataset that can be used as example for 2x2 factorial designs. [Internet]. 2017. Available from: <http://bioconductor.org/packages/release/data/experiment/html/estrogen.html>
66. Adam E, Janes J, Lowney R, Lambert J, Thampi P, Stromberg A, et al. Comparison of Chondrogenic Differentiation Potential Between Different Adult and Fetal Cell Types. Vet Surg [Internet]. 2019;48(3):375–87. Available from: <https://doi.org/10.1111/vsu.13183>
67. Li H jun, Wang L ya, Qu H na, Yu L hua, Burnstock G, Ni X, et al. P2Y2 receptor-mediated modulation of estrogen-induced proliferation of breast cancer cells. Mol Cell Endocrinol [Internet]. 2011;338(1–2):28–37. Available from: <http://dx.doi.org/10.1016/j.mce.2011.02.014>
68. Storey VC. Understanding semantic relationships. VLDB J. 1993;2(4):455–88.
69. Carlson M. GO.db: A set of annotation maps describing the entire Gene Ontology [Internet]. 2016. Available from: <https://bioc.ism.ac.jp/packages/3.3/data/annotation/html/GO.db.html>

70. Götz S, García-Gómez JM, Terol J, Williams TD, Nagaraj SH, Nueda MJ, et al. High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Res.* 2008;36(10):3420–35.
71. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* 2007. 2010;25(11).
72. Dixon CJ, Bowler WB, Fleetwood P, Ginty AF, Gallagher JA, Carron JA. Extracellular nucleotides stimulate proliferation in MCF-7 breast cancer cells via P2-purinoceptors. *Br J Cancer.* 1997;75(1):34–9.
73. Wagstaff SC, Bowler WB, Gallagher JA, Hipkind RA. Extracellular ATP activates multiple signalling pathways and potentiates growth factor-induced c-fos gene expression in MCF-7 breast cancer cells. *Carcinogenesis* [Internet]. 2000;21(12):2175–81. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/11133806>
74. Jin H, Eun SY, Lee JS, Park SW, Lee JH, Chang KC, et al. P2Y2 receptor activation by nucleotides released from highly metastatic breast cancer cells increases tumor growth and invasion via crosstalk with endothelial cells. *Breast Cancer Res* [Internet]. 2014;16(5):R77. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4406012&tool=pmcentrez&rendertype=abstract>
75. Moseley HNB. Error Analysis and Propagation in Metabolomics Data Analysis. *Comput Struct Biotechnol J* [Internet]. 2013;4(5):e201301006. Available from: <http://dx.doi.org/10.5936/csbj.201301006>
76. Hinderer EW, Flight RM, Moseley HNB. GOcats: A tool for categorizing Gene Ontology into subgraphs of user-defined concepts. *bioRxiv.* 2018;08.
77. Barrell D, Dimmer E, Huntley RP, Binns D, O'Donovan C, Apweiler R. The GOA database in 2009 - An integrated Gene Ontology Annotation resource. *Nucleic Acids Res.* 2009;37(SUPPL. 1):396–403.
78. McCarthy FM, Wang N, Magee GB, Nanduri B, Lawrence ML, Camon EB, et al. AgBase: A functional genomics resource for agriculture. *BMC Genomics.* 2006;7:1–13.

79. Adam EN. Differential Gene Expression in Equine Cartilaginous Tissues and Induced Chondrocytes. ProQuest Diss Theses [Internet]. 2016;152. Available from: <https://search.proquest.com/docview/1990661674?accountid=172684>
80. Mienaltowski MJ, Huang L, Frisbie DD, McIlwraith CW, Stromberg AJ, Bathke AC, et al. Transcriptional profiling differences for articular cartilage and repair tissue in equine joint surface lesions. BMC Med Genomics. 2009;2:1–14.
81. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for Illumina Sequence Data. Bioinformatics [Internet]. 2014;btu170. Available from: <http://www.usadellab.org/cms/?page=trimmomatic>
82. Wang K, Singh D, Zeng Z, Coleman SJ, Huang Y, Savich GL, et al. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. Nucleic Acids Res. 2010;38(18):1–14.
83. Hinderer EW, Moseley HNB. Equine Enrichment Results Directory from GOcats Annotation Enrichment. 2018; Available from: <https://figshare.com/s/73b8f454516f0e7cbec7>
84. Howlader N, Noone A, Krapcho M, Garshell J, Miller D, Altekruse S, et al. SEER Cancer Statistics Review, 1975-2012 [Internet]. Bethesda, MD; 2015. Available from: [https://seer.cancer.gov/archive/csr/1975\\_2012/](https://seer.cancer.gov/archive/csr/1975_2012/)
85. Association AL. Kentucky Lung Cancer Rates [Internet]. 2019. Available from: <https://www.lung.org/our-initiatives/research/monitoring-trends-in-lung-disease/state-of-lung-cancer/states/KY.html>
86. Liu J, Murali T, Yu T, Liu C, Sivakumaran TA, Moseley HNB, et al. Characterization of squamous cell lung cancers from Appalachian Kentucky. Cancer Epidemiol Biomarkers Prev. 2019;28(2):348–56.
87. Huang Q, Tan Q, Mao K, Yang G, Ma G, Luo P, et al. The role of adrenergic receptors in lung cancer. Am J Cancer Res [Internet]. 2018;8(11):2227–37. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/30555740> <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6291649>
88. Hinderer EW, Moseley HNB. KLCG-TCGA MutSig Enrichment Enrichment Results. 2019; Available from: <https://figshare.com/s/4a43a21193c7e99013fc>

89. Bird S, Klein E, Loper E. Natural Language Processing with Python. O'Reilly Media; 2009.
90. Powers DMW. Applications and Explanations of Zipf's Law. In: New Methods in Language Processing and Computational Natural Language Learning. Sydney; 1998. p. 151–60.
91. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9(3):99–104.

## VITA

### Education

- 2010-2014     Bellarmine University, Louisville, KY  
                  B.Sc. *Magna cum laude*, Biochemistry and Molecular Biology  
                  GPA: 3.83/4.00
- 2014-2019     University of Kentucky, Lexington, KY  
                  Ph.D. Molecular and Cellular Biochemistry  
                  GPA: 4.00/4.00

### Research Experience

- *Biomedical Ontology Specialist, BigR.io* .....2019-present
- *Graduate Research Assistant, University of Kentucky*  
*Advisor: Dr. Hunter N.B. Moseley*.....2015-present
- *Graduate Research Assistant, University of Kentucky/Transposagen Biopharmaceuticals*  
*Advisor: Dr. Carlisle Landel*..... Jan 2015-Mar 2015
- *Graduate Research Assistant, University of Kentucky*  
*Advisor: Dr. Matthew Gentry*..... Oct 2014-Dec 2014
- *Undergraduate Part-Time Research Associate, University of Louisville*  
*Advisor: Dr. Hunter N.B. Moseley*..... 2011-2014

### Honors and Awards

- *University of Kentucky enrollment incentive* ..... 2014

\$3000 awarded for outstanding prospective students

- *Bellarmino Monsignor Horrigan Scholar Scholarship*..... 2010-2014

\$47,500 awarded over 4 years as an academic scholarship

- *Bellarmino University Dean's List*..... 2011-2014

Eight semesters of inclusion for top-percentile GPA

- *Presidential Bellarmine University Achievement* ..... 2011-2014

\$3,000 awarded over 3 years

- *Research Experiences for Undergraduates summer program* ..... 2012

\$5,000-Ten-week competitive stipend

### Publications

1. **Eugene W. Hinderer III**, Robert M. Flight, Rashmi Dubey, James N. MacLeod, and Hunter N.B. Moseley. "Advances in Gene Ontology Utilization Improve Statistical Power of Annotation Enrichment" PLOS One. 2019 (accepted).
2. **Eugene W. Hinderer III** and Hunter N.B. Moseley. "GOcats: A tool for categorizing Gene Ontology into subgraphs of user-defined concepts" BioRxiv preprint 306936 (2018).
3. Rashmie Abeysinghe, Fengbo Zheng, **Eugene W. Hinderer III**, Hunter N.B. Moseley, and Licong Cui. "A Lexical Approach to Identifying Subtype Inconsistencies in Biomedical Terminologies" Quality Assurance of Biological and Biomedical Ontologies and Terminologies Workshop -- Bioinformatics and Biomedicine (BIBM), 2018 IEEE International Conference S12204. 2018.
4. Rashmie Abeysinghe, **Eugene W. Hinderer III**, Hunter N.B. Moseley, and Licong Cui. "Auditing Subtype Inconsistencies among Gene Ontology Concepts" The 2nd

International Workshop on Semantics-Powered Data Analytics (SEPDA 2017) --  
Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference  
1242-1245. 2017.