GOcats: A tool for categorizing Gene Ontology into subgraphs of user-defined concepts

Eugene W. Hinderer III[1], Robert M. Flight[2,3], and Hunter N.B. Moseley[1, 2, 3, 4]

[1]Department of Molecular and Cellular Biochemistry, [2]Markey Cancer Center,

[3]Center for Environmental and Systems Biochemistry, [4]Institute for Biomedical Informatics, University of Kentucky, Lexington, KY, 40536-0093, USA

Running Title: GOcats

Keywords: gene ontology, ontological scoping, semantic correspondence, emergent concepts, annotation enrichment analysis

Correspondence to: hunter.moseley@uky.edu

Software and full results available at:
http://software.cesb.uky.edu
https://figshare.com/s/26b336a06946a9248e08 (software version 1.0.0c)
https://github.com/MoseleyBioinformaticsLab/GOcats (software version 1.1.4)
https://pypi.python.org/pypi/GOcats (software version 1.1.4)
http://gocats.readthedocs.io/en/latest/index.html (documentation)
https://figshare.com/s/cc1abe7e2e5c4ae09500 (results)

Abstract

Ontologies are used extensively in scientific knowledgebases and repositories to organize the wealth of available biological information. However, gene-annotation enrichment queries utilizing these resources can provide thousands of results with weak statistical significance that may be difficult to interpret without manually sorting into higher-order categories. Additionally, some ontology relations are problematic with respect to scope and hamper categorization, necessitating their omission lest erroneous term mappings occur. This omission leads to at least a 6% reduction in retrievable relational information in the Gene Ontology, yet including these terms results in over 31% (325180 out of 1036141) of term mappings being erroneous with respect to categorization when current tools are used.

To address these issues, we present GOcats, a novel tool that organizes the Gene Ontology (GO) into subgraphs representing user-defined concepts, while ensuring that all appropriate relations are congruent with respect to scoping semantics. We tested GOcats performance using subcellular location categories to mine annotations from GO-utilizing knowledgebases and evaluating their accuracy against immunohistochemistry datasets in the Human Protein Atlas (HPA). In comparison to mappings generated from UniProt's controlled vocabulary and from GO slims via OWLTools' Map2Slim, GOcats outperforms these methods without reliance on a human-curated set of GO terms. By identifying and properly defining relations with respect to semantic scope, GOcats can use traditionally problematic relations without encountering erroneous term mapping. We then applied GOcats in the comparison of HPA-sourced knowledgebase annotations to experimentally-derived annotations provided by HPA directly. During the comparison, GOcats improved correspondence between the annotation sources by adjusting semantic granularity. Utilized in this way, GOcats can perform an accurate knowledgebase-level evaluation of curated HPA-based annotations.

Finally, we show that GOcats' unique handling of relations improves enrichment statistics over conventional methods by integrating GOcats into the categoryCompare2 enrichment analysis pipeline and performing enrichment on a publicly-available breast cancer dataset. Specifically, we saw significant improvement (one-sided binomial test p-value=1.86E-25) in 182 of 217 significantly enriched GO terms identified from the conventional method when GOcats' path traversal was used. We also found unique, significantly enriched terms using GOcats, whose biological relevancy has been experimentally demonstrated elsewhere.

Author Summary

We present the Gene Ontology Categorization Suite or GOcats which is designed, in part, to help scientists interpret large-scale experimental results by organizing redundant and highly-specific annotations into customizable, biologically-relevant concept categories. Ontologies like Gene Ontology organize the language of a discipline, e.g. molecular and cellular biology, and provide a standardized and consistent context for annotation terms, i.e. descriptions of domain-specific concepts. During GOcats' development, we encountered a pervasive issue related to how part-whole (mereological) relations are handled by ontology categorization methods. We report that this issue either limits the amount of retrievable information available from ontologies, or results in erroneous categorizations depending on whether or not these

relations are included in the analysis. Therefore, we developed a custom graph parsing scheme that allows these relations to be incorporated without resulting in erroneous term categorizations. This enables a more robust semantic scoping utilization of Gene Ontology, facilitating summarization of annotations in large data sets. We also demonstrate that GOcats is well suited for comparing results from separate annotation sources due to its ability to allow adjustment of categories to the appropriate annotation term granularity. GOcats thus facilitates more robust interpretation and comparison of experimental and knowledgebase annotation sources and provides new tools for semantic scoping utilization and development of Gene Ontology. Furthermore, when used alongside annotation enrichment tools such as categoryCompare2, GOcats' unique method for inferring category membership results in both improved enrichment statistics and the identification of enriched terms otherwise impossible-to-identify with statistical significance when compared to using conventional ontology inference rules.

Introduction

Background

Biological and biomedical ontologies such as Gene Ontology (GO) (Ashburner *et al*, 2000) are indispensable tools for systematically annotating genes and gene products using a consistent set of annotation terms. Ontologies 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. These ontologies are created, maintained, and extended by experts with the goal of providing a unified annotation scheme that is readable by humans and machines (Smith *et al*, 2007).

Ontologies and other controlled vocabularies (CV) within the biomedical domain such as GO and the Unified Medical Language System (Gene Ontology consortium, 2015; Bodenreider, 2004) 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 was, and still 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, as large-scale and high-throughput investigations continue to advance, ontologies are also evolving as they are used in ways that extend beyond their initial purpose as annotation reference utilities. Ontology annotations are utilized alongside automated pipelines that analyze protein-protein interaction networks and form predictions of unknown protein function based on these networks (Veres *et al*, 2015; Huttlin *et al*, 2015), for gene function enrichment analyses, and are now being leveraged for the creation of predictive disease models in the scope of systems biochemistry (Papatheodorou *et al*, 2015).

Ontologies and omics-level research

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 (Subramanian *et al*, 2005). However, such studies often result in many individually enriched GO terms that are highly specific and weakly justified by statistical significance. Often times, the resulting large sets of weakly enriched terms are difficult to interpret without manually sorting into appropriate descriptive categories (Na *et al*, 2014). 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 (Veres *et al*, 2015). While GO provides a way to annotate large numbers of genes and gene products with high semantic specificity, it cannot identify emergent concepts from GO organized as categories of GO terms. 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. These category-defining terms enable a more robust and easily interpretable organization of genes and gene products for the investigation of specific biological and disease processes and facilitate the development of complex biological models such as bio-macromolecular interaction and metabolic networks.

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 (Uhlen *et al*, 2015). Such efforts are creating a huge amount of omics-level experimental data that is being 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.

Anatomy of the Gene Ontology

The GO database itself represents a CV of biological, and biochemical terms that are each assigned a unique alphanumeric code, which is used to annotate genes and gene products in many other databases, including UniProt (The UniProt Consortium, 2015a) and Ensembl (Cunningham *et al*, 2015). The ontology is divided into three sub-ontologies: Cellular Component, Molecular Function, and Biological Process. Each can be envisioned as a graph or network where terms are nodes connected by edges that describe how each term relates to one another. For example, the term “DNA methylation” (GO:0006306) is connected to the term “macromolecule methylation” (GO:0043414) by the is_a relation. In this case, ontological terminology defines the term “macromolecule methylation” as a “parent” of the term “DNA methylation.” The three sub-ontologies mentioned are “is_a disjoint” meaning that there are no is_a relations connecting any node among the three 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 similar to 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 part-whole (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.

There are three versions of the GO database, each containing aspects of the CV with varying complexity: *go-basic* is filtered to exclude relations that span across multiple sub-ontologies and to include only relations that point toward the root of the ontology; *go* or *go-core* contains additional relations, such as has_part that may span sub-ontologies and which point both toward and away from the root 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 (ChEBI) ontology (Munoz-Torres & Carbon, 2017). 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.

Categorization-relevant 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; 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 (Noy & Wallace, 2005; Gene Ontology consortium, 2017). 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 edges are removed in the graph. For those interested in organizing 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.

Term categorization approaches

Issues of term organization and term filtering have led to the development of GO slims—manually cut-down versions of the gene ontology containing only generalized terms (GO Slim and Subset Guide) which represent concepts within GO, as well as other software, like Categorizer (Na *et al*, 2014), which 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), or GOATools (OWLTools, 2015; Tang *et al*), 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 (Binns *et al*, 2009), using Map2Slim 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 (The UniProt Consortium, 2015b).

However, it is smaller and less maintained than GO slims, and is intended for use only within UniProt's native data structure.

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 of (Na *et al*, 2014; Jiang, 1997; Lin, 1989; Resnik, 1999; Schlicker *et al*, 2006). 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 (Na *et al*, 2014). 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 will not be 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 would only cause 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.

Moreover, current methods for utilizing ontological annotations—such as in the enrichment studies mentioned previously—often rely on manual intervention for sorting annotations into biologically meaningful categories either directly, or indirectly through the use of GO slims (Veres *et al*, 2015). This categorization of annotations typically occur after enrichment analysis has been performed (Na *et al*, 2014). But the limitations of these methods often burden researchers with manual inspection of both fine-grained and course terms, potentially introducing human error and discrepancies into analyses, hindering reproducibility, complicating interpretation, and ultimately impacting the statistical power of gene and GO set enrichment analyses.

Axiomatic versus semantic scoping interpretation and use of mereological relations in GO

While ensuring mereological universality in relation associations using current axioms is important within the purview of ontology development, 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.

As mentioned, the current axiomatic representation of mereological relations requires the use of ontology versions which lack certain relations, resulting in a loss of retrievable information. If has_part edges—which point toward terms of narrower scope—were to be inversed 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.

Therefore, 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 very 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.

Emergent concepts

Emergent properties of a complex system are those which arise from individual components working in concert to perform new functions that were not possible by the individual components themselves. In the same respect, an emergent concept arises from the interplay of existing or predefined concepts. In the context of ontologies, terminology for concepts are explicitly defined within the nodes of the ontological graph. We believe that non-explicit, emergent concepts may be discovered in ontologies by evaluating the intersection of two or more broad-topic, concept-centric ontological subgraphs. For example, one may define an emergent concept of “nuclease activity involved in autophagy” by first categorizing terminology into the concepts “nuclease activity” and “autophagy” and then assigning the intersection of these to this new emergent concept. While the hierarchical nature of ontologies makes this logic obvious, in practice, identifying emergent concepts is non-trivial. This is because ontologies, by design, do not make a distinction between broad-topic-level concepts like autophagy and more fine-grained, detail-level concepts like "engulfment of target by autophagosome." This makes sense because such distinctions would be arbitrary and would vary depending on context. As mentioned, ambiguities in terms of scope among relations further complicate the generalization of concept-centric subgraphs toward this end. Therefore, we anticipate that customizable, extra-ontological tools will be needed to identify these emergent concepts.

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 revising, especially as ontologies evolve or merge. A recent review of current methods for biomedical ontology mapping highlights the importance in developing semi-automatic methods to aid in ontology evolution efforts and reiterates the aforementioned concept of semantic correspondence in terms of scoping between terms (Groß *et al*, 2016). 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 (Groß *et al*, 2013; Cesar *et al*, 2013). Ontology merging also continues to be an active area of development for integrating functional, locational, and phenotypic information. To aid in this endeavor, another recent review points out that it is crucial to integrate phenotypic information across various levels of organismal complexity, from the cellular level to the organ system level (Papatheodorou *et al*, 2015). Thus, organizing location-relevant ontology terms into discrete categories is an important step toward this end.

GO Categorization Suite (GOcats)

For the reasons stated above, we have developed a new 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 mereological relation edges within GO.

Here, we demonstrate the utility of GOcats and the effectiveness of evaluating mereological relations with respect to semantic scope by categorizing the GO Cellular Component ontology into broad-level concepts representing cellular components. We used the concept-centric subgraphs produced by GOcats to create a mapping of fine-grained terms to their chosen concept-representative term. Using these mappings, we categorized knowledgebase-derived gene annotations and compared this automated categorization to publicly available datasets of manually-categorized gene annotations assigned by researchers at the HPA following immunohistochemistry experiments.

Furthermore, we illuminate the extent of information loss or potential for misinterpretation of has_part relations in their current form if they are excluded or included in current GO term categorization methods, respectively. Finally, we demonstrate that GOcats' reinterpretation of has_part can retain all information from GO while drawing appropriate categorical inferences for the purpose of annotation enrichment. This reinterpretation has the added benefit of improving the statistical power of annotation enrichment analyses.

Design and Implementation

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 separate ontologies. 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 relation contained therein, nor are the additional database cross-references meaningful to this study.

While *go-basic* is a true DAG, *go-core* is not strictly acyclic due to its additional has_part relations. However, when we inversed traversal of has_part into the part_of_some interpretation, acyclicity was maintained. Therefore, we refer to our *go-core* graph as a DAG (see below).

GOcats is a Python package written in version 3.4.2 of the Python program language (van Rossum & Drake, 2011). It uses a Visitor design pattern implementation (Gamma *et al*, 1994) to parse the *go-core* Ontology database file (Gene Ontology consortium, 2015). The DAG hierarchal structure of the ontology is represented as a graph implemented using customized Python objects. Searching with user-specified sets of keywords for each category, GOcats extracts subgraphs of the GO DAG (sub-DAGs) and identifies a representative node for each category in question and whose child nodes are detailed features of the components (Figure 1a).

Figure 2 illustrates this approach in more detail. The user-provided keyword sets are used by GOcats to query GO terms' name and definition fields to create an initial seeding of the sub-DAG with terms that contain at least one keyword, this seeding is a list of nodes from the whole GO graph (supergraph) that pass the query.

```
FOR node in supergraph.nodes
    IF keyword from keyword_list in node.name or node.definition
        APPEND node to subdag.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 node in subgraph.nodes
    LOOKUP child_node AND parent_node from subgraph.edges
    ADD child_node to node.child_node_set IF node == /
    edge.parent_node
    ADD parent_node to node.parent_node_set IF node == /
    edge.child_node
```

GOcats then selects a category representative node to represent the sub-DAG. To do this, a list of candidate representative nodes is compiled from non-leaf nodes, i.e. root-nodes in the sub-DAG 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 node in subgraph
    IF node.child_node_set != None AND ANY keyword in node.name
        APPEND node in subgraph.nodes to candidates
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 may not contain a keyword in its name or definition but still may be part of the sub-DAG 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. (Figure 2, and see below).

```
Comprehensive extension:
FOR node in supergraph
    IF ANY (node in node.ancestors) in subgraph
```

```
        APPEND node to subgraph.nodes
UPDATE subgraph # appropriate edges added and parent/child nodes
                # assigned
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 ⍰ /
    supergraph.end_node.descendents
        APPEND node to subgraph.nodes
UPDATE subgraph # appropriate edges added and parent/child nodes
                # assigned
```

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.

To overcome the previously mentioned issues regarding scoping ambiguity among mereological relations, we manually 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 2, Figure 3). 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.

For mapping purposes, Python dictionaries are created which map GO terms to their corresponding category or categories. For inter-sub-DAG analysis, another Python dictionary is created which maps each category to a list of all its graph members. By default, fine-grained terms do not map to category root-nodes that define a sub-DAG that is a superset of a category with a root-node nearer to the term. For example, a member of the “nucleolus” sub-DAG would map only to “nucleolus,” and not to both “nucleolus” and “nucleus”. However, the user has the option to override this functionality if needed. Mapping supersets is a requirement for visualizing concept membership in graph representations using tools like Cytoscape (Figure 4).

Results

GOcats compactly organizes GO subcellular localization terms into user-specified categories

GOcats utilizes the DAG structure of GO along with a small number of user-specified keywords and/or GO terms to extract an arbitrary number of subgraphs from GO, each representing a broad category with which fine-grained GO terms can be mapped to the root-node of each subgraph, without needing explicitly-defined GO terms (Figures 1 and 2). We evaluated the automatic extraction and categorization of 25 subcellular locations, using the “comprehensive” method of subgraph extension (Figure 2, see methods). Of these, 22 contained a designated GO term root-node that exactly matched the concept intended at the creation of the keyword list (Table 2). These subgraphs account for approximately 89% of GO's cellular compartment ontology. Note that because subgraph nodes may root to more than one representative root node, the totals in Table 2 do not add up to the total number of GO terms in Cellular Component.

While keyword querying of GO provided an initial seeding of the growing subgraph, we also emphasize the necessity of re-analyzing the GO graph to find terms missed by the keyword search, to remove terms erroneously added by the keyword search, and to add appropriate subgraph terms not captured by the keyword search. In table 2, this is apparent by comparing the number of seeded nodes from the keyword search to the total nodes, and by the number of nodes added during the extension of the subgraph. For example, the "cytoplasm" subgraph grew from its initial seeding of 296 nodes to 1197 nodes after extension. Conversely, while 136 nodes were seeded by keyword for the "bacterial" subgraph, only 16 were truly rooted to the representative node, and necessitated the removal of serendipitously added nodes.

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 and the remaining are rooted only to macromolecular complex and another compartment that was extracted.

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 (Figure 4a). The amorphous clustering of nodes immediately surrounding macromolecular complex represents those terms which root only to macromolecular complex and the highly clustered circular nodes represent groupings which root to the same root-node(s). The patterns of connectedness in this network make sense biologically, within the constraints of GO's internal organization. For example, intracellular organelles tend to be clustered about cytoplasm, with the exception of nucleus which the GO consortium does not consider to be part of the cytoplasm. But the macromolecular complex category somewhat complicates the visualization of category organization within GO which indicates just how large and interconnected this category is in 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 4b). 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.

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.

To assess the accuracy of GOcats' category subgraph contents, we evaluated the similarity of these subgraph contents to subgraphs of the manually-curated UniProt subcellular localization CV (The UniProt Consortium, 2015b, 2015a) as illustrated in Figure 5. In comparing the overlap of terms between UniProt's CV and corresponding category-representative nodes produced by GOcats, most GOcats-derived subgraphs are large supersets of UniProt subgraphs. Table 3 shows that 12 of the GOcats-derived compartments had identical root nodes in UniProt's considerably smaller controlled vocabulary. Of these, 6 contained 100% inclusion and were approximately 20 times larger on average. The others contained between 56.2% and 84.6% inclusion. Some discrepancies in the organizational patterns between UniProt and GO may account for the lower inclusion. One major discrepancy is UniProt's organization of the plasma membrane and other cellular envelopes.

We also performed comparisons with subgraphs created by M2S. This method is more comparable to GOcats, because it directly utilizes the GO graph structure. In comparing the category subgraphs created by GOcats and M2S, the mappings for most categories are in very close agreement, as evidenced by both high inclusion and Jaccard indices in Table 4 and further highlighted in Figures 6a, 6b and Supplemental Data 1a-v. However, in some categories, M2S and GOcats are in disagreement as illustrated in Figure 6c and Supplemental Data 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 theses 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 Data 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. Despite the fact that the majority of 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. We suspected that the differences in mapping could be due to our reevaluation of the has_part edges with respect to scope. As shown in Table 4, 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.

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 (see Methods, Equations 2-6). As shown in Table 5, there are 23,640 pM_Fs in Cellular Component, 8,328 pM_Fs in Molecular Function, and 89,815 pM_Fs in Biological Process. Comparatively, the amount of pM_Fs is 42%, 13%, and 16% the size of the M_T, in Cellular Component, Molecular Function, and Biological Process, respectively.

The conventional solution to avoid these errors are to use versions of ontologies that remove edges like has_part. (Binns *et al*, 2009). 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_T 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 5, 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 here 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 erroneously. 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 M2S, which evaluates non-scoping ontology relations as if they describe scoping semantics. For example, terms linked by an active relation like *regulates* 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 (see Methods, Equations 7-8). 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 ($M_{pair,M2S}$) since they represent a problematic evaluation of scoping semantics. This contrasted with 710,961 correct mappings that intersected the GOcats mapping pairs ($M_{pair,GOcats}$) 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 6).

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. However, we intentionally utilized the full GO-core ontology to illustrate the danger in using tools that do not provide explicit semantic control on how ontologies are utilized. Furthermore, tools that can semantically utilize ontological information to a fuller extent while providing accurate categorization of terms represent an advancement and will be instrumental for improving annotation enrichment analyses.

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. Using the set of GO terms annotated in the HPA's immunohistochemistry localization raw data as "concepts" (Table 7), 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 (Binns *et al*, 2009). These annotation-category mappings were also visualized using Cytoscape (Figure 4C). Next, we evaluated how these derived annotation categories matched raw HPA data GO annotations. Figure 7 illustrates the data analysis steps utilized in this evaluation. 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 8a). 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 5.971 seconds when filtered to the cellular localization sub-ontology and 9.248 seconds when unfiltered, while M2S performed its mapping on the same data in 13.393 seconds. Although comparable, GOcats should offer appreciable computational improvement on significantly larger datasets. This is rather surprising since GOcats is implemented in Python (van Rossum & Drake, 2011), an interpreted language, versus M2S which is implemented in Java and compiled to Java byte code.

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 8). In doing so, GOcats is able to 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 8b). There is not an appreciable change in the number of gene annotations not found in the knowledgebase (data not shown). By enabling users to quickly customize the level of semantic specificity for annotation categories, a more meaningful query of knowledgebases annotations is possible with GOcats.

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. Due to the UniProt CV's poor performance in the previous results, we omitted it from this evaluation. 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 9). 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 that disagreements were present across nearly all categories (Table 9).

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 9, 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 more 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 immediately 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 9 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.

GOcats’ reinterpretation of has_part relations results in improved enrichment statistics.

We incorporated GOcats-derived ontology ancestor paths (paths from fine-grained terms to more general, categorical terms) into the categoryCompare (Flight *et al*, 2014) annotation enrichment analysis pipeline and performed annotation enrichment on an Affymetrix microarray dataset of ER+ breast cancer cells with and without estrogen exposure(Huber & Gentleman, 2017). 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.

Assessment of p-values from significantly enriched terms using GOcats' paths versus the traditional method of omitting has_part edges shows that GOcats reliably improves the statistical significance of term enrichment results through its unique re-interpretation of relation semantics (Figure 10, Supplemental Data 3). 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 (one-sided binomial test p=1.86E-25).

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 (Supplemental Data 4). Four of these terms involve purinergic nucleotide receptor activity, which has been implicated elsewhere in other investigations related to breast cancer (Jin *et al*, 2014). GOcats-augmented ontology paths therefore show promise in both allowing for additional information to be retrieved from annotation enrichment analyses and for improving the statistical power of enrichment.

Discussion

In this study, we: i) demonstrated the increase in retrievable ontological information content via reevaluating mereological relations to make them congruent with respect to semantic scope, ii) applied our new method GOcats toward the categorization and utilization of the GO Cellular Component sub-ontology, iii) evaluated the ability of GOcats and other mapping tools to relate HPA experimental to HPA knowledgebase GO Cellular Component annotation sources and iv) demonstrated some improvements afforded by GOcats toward annotation enrichment experiments. Our results indicate that, when compared to UniProt's CV, GOcats' mapping was able to assign gene annotations from the UniProt and Ensembl database to subcellular locations with greater accuracy when compared to a raw dataset of gene localization annotations. It is important to note that UniProt's CV was not intended to be used as a method to categorize gene annotations. Nevertheless, it is itself a DAG with a structure comparable to GO and we analyzed the graph and mapped fine-grained terms to general terms using the same techniques used for GO. Moreover, GOcats comparison to M2S demonstrates similar mapping performance between the two methods, but with GOcats providing important improvements in mapping, speed, ease of use, and flexibility of use. Using GOcats, the user can 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. We have used this functionality to automatically map gene annotations from Ensembl and UniProt-GOA knowledgebases and compared these localization assignments to manually-assigned localizations taken from high-throughput immunohistochemistry experiments performed by the HPA [13]. We show that GOcats allows a robust organization of Cellular Component into user-specified categories, while providing more automation than current methods. Furthermore, we demonstrate GOcats' ability to query and organize gene-specific annotations from knowledgebases into experimentally-verified general subcellular locations with safer scoping utilization over other methods that require specific versions of GO. We demonstrate GOcats' utility for evaluating annotation assignment consistency between raw experimental data and knowledgebase data, highlighting this software's promise for knowledgebase curation and quality control. Finally, we showed that GOcats' improves the statistical power of annotation enrichment analyses, enabling the detection of statistically significant enriched annotations that would otherwise be missed due to the loss of information from excluded has_part relations.

Issues with other methods

Other methods used to summarize or categorize GO terms into biologically relevant concepts: i) rely heavily on static and manually-maintained GO slims, ii) require the user to create GO slims by hand-selecting GO terms and creating a GO slim from scratch, or iii) require the user to perform post-analysis categorization of GO terms, typically enriched GO terms from gene enrichment studies. Caveats of such methods include burdening the user with the tasks of finding and editing, or even creating the appropriate GO slims to suit their research, limiting categorization to only those concepts that are explicitly defined in GO, and in the case of post-enrichment categorization tools, limiting the statistical power of the analysis by not automatically binning gene annotations into categories prior to enrichment. Until GOcats, there has been no resource developed which categorizes GO terms into subsets representing concepts without user-specification of individual GO terms or the use of GO slims, which operates using only the "expected" DAG structure of GO.

M2S is one widely-utilized GO term categorization method that is available as part of the OWLTools Java application (OWLTools, 2015). The Perl version of M2S has been integrated into the Blast2GO suite since 2008 (Götz *et al*, 2008; Chris Mungall

BDGP, 2013) and this gene function annotation tool has been cited in over 1500 peer-reviewed research articles (Google Scholar as of Nov. 28, 2017). We tested to ensure that the Perl and Java versions of M2S produced the GO term mappings for a given dataset and GO slim, and therefore had the same mapping errors (see Methods, Supplemental Data 5). Although the number of pM$_F$s 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. This can be appreciated in the disagreements among the subgraphs created by M2S and GOcats, particularly in the “plasma membrane” category. Here, M2S provides results which imply that concepts such as “cell projection cytoplasm” are a sub-concept of the term “plasma membrane” (Figure 3). Limiting information content to avoid this issue represents a significant limitation to enrichment tools that naively utilize the ontology structure to categorize enrichment terms. We designed GOcats to properly incorporate scoping edges that are otherwise missed when categorizing GO. Since we show that M2S encounters issues with categorization unless a scoping-safe ontology is used (i.e. limited to is_a and part_of relations), improvements from GOcats could offer far reaching effects.

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 (Storey, 1993). This includes concepts of part-whole (mereology), general-specific (hyponymy), feature-event, time-space (i.e spaciotemporal 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 are 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 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 (Binns *et al*, 2009). Furthermore, Bioconductor's GO.db (Carlson, 2016) 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 avoid scope-specific mapping errors, it also limits the amount 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 5). 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 appreciable 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 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 prevalence of additional, non-scope 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. GOcats represents a step toward a more thorough evaluation of the semantics contained within ontologies by handling relations differently according to the linguistic correspondences that they represent. In the case of relations such as *has_part,* this involves augmenting the correspondence directionality when it is appropriate for the task at hand, which is to organize terms into categories. As a prototypical proof-of-concept, we classified the is_a, has_part, and part_of relations into a common "scoping" correspondence type and manually 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 unlikely misinterpretation of universality in this strategy is preferable to the loss of information experienced when using cut-down 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 simple 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 manual categorization of semantic correspondences through the development of heuristic methods that identify and categorize these among the hundreds of relations in the Relations Ontology (Relations Ontology, 2016; Smith *et al*, 2007). As a good starting point, we suggest using five general categories of relational correspondence for reducing ambiguity (Table 1): scope (hyponym-hypernym), mereological, a subclass of scope (meronym-holonym), spatiotemporal (process-process, process-entity, entity-entity), active (actor-subject), and other.

Organization of Cellular Component with respect to “macromolecular complex”

Our analysis of the overall structure of Cellular Component revealed some unexpected properties. As mentioned previously, the category "macromolecular complex" accounted for nearly two-thirds of the entire Cellular Component (2,083 terms), with a moderate portion (~25%) only being rooted to "macromolecular complex" as opposed to other sensible root locations such as "cytosol," "cytoplasm," or "extracellular." Upon closer inspection, some terms rooted only to "macromolecular complex" contained implicit or explicit indications of being associated or contained within other subcellular locations in their definition lines but had no relation with the location in question in the GO graph. For example, "Seh1-associated complex" is defined as "A protein complex that associates dynamically with the vacuolar membrane, and is proposed to have a role in membrane-associated trafficking or regulatory processes…" However, within the DAG this term has only one parent, "Protein complex" which has one parent, "macromolecular complex." Though it may be true that the annotators intentionally root some terms conservatively, avoiding a part_of relation when a complex dynamically associates with a compartment, it is unclear to us why this example and others like it do not contain another root such as "cytosol" or "cytoplasm." Nevertheless, GOcats allows for the detection of such terms, which may be of use to GO curators and others interested in evaluating the structural organization of GO.

Using GOcats to handle differences in semantic granularity

As our results indicate, 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. As we show, utilizing only the set of specific annotations used in

HPA's experimental data, M2S's mapping matches only 366 identical gene annotations from the knowledgebase, which is similar to when GOcats is provided with categories matching that set (Figure 8a). GOcats solves this problem by allowing researchers to easily 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 8b).

Using GOcats for curation and quality control

As GO continues to grow, automated methods to evaluate the structural organization of data will become necessary for curation and quality control. For instance, we recently collaborated in the evaluation of a method to automate auditing of potential subtype inconsistencies among terns in GO (Abeysinghe *et al*, 2017). Because GOcats allows versatile interpretation of the GO 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 allow researchers to check the integrity 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 protein atlas, and v) one or more independent antibodies yielding staining patterns not consistent with experimental gene/protein characterization data, but supported by siRNA assay (Uhlen *et al*, 2015; Data Quality Assurance and Scoring, 2016). 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” (Uhlen *et al*, 2015; Assays and Annotation, 2016). 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.

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.

In our evaluation of enrichment results comparing GOcats ancestor paths to traditional GO ancestor paths in the enrichment analysis of a publicly-available breast cancer dataset, we demonstrate a highly statistically significant improvement (p=1.86E-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. Moreover, we 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 < 0.002 with FDR set to 0.05) of four terms related to purinergic nucleotide receptor signaling which has been implicated in predicting breast cancer metastasis in other studies (Jin *et al*, 2014). Fundamentally, the addition of part_of_some interpretation of has_part relations provides additional annotation information that can be aggregated in additive manner during annotation enrichment analysis, preventing the misinterpretation of part_of_some relations. In turn, the additional annotation information improves the statistical power of the annotation enrichment analysis, allowing the detection of additional enriched annotations with statistical significance from the same dataset.

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, lowering the bar for extracting useful information from exponentially growing scientific knowledgebases and repositories in a semantically safer manner. GOcats is a versatile software tool applicable to data mining, annotation enrichment analyses, ontology quality control, and knowledgebase-level evaluation and curation.

Materials and Methods

Creation and visualization of generalized GO Cellular Component categories

We provided GOcats with the GO graph, data-version: releases/2016-01-12 and sets of keywords representing each subcellular-specific location identified in HPA's high-throughput immunohistochemistry experiments (Table 7), in addition to other pertinent localizations (Table 2). To assess the relative size and structure of subgraphs within GO, we visualized the category subgraphs as a network using Cytoscape 3.0 (Shannon *et al*, 2003) (Figure 4a-c). GOcats outputs a dictionary of individual GO term keys with a list of category-defining root-node values as part of its normal functionality.

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 (The UniProt Consortium, 2015a) for comparison to GOcats. To accomplish this, we parsed and recreated the graph structure of UniProt's subcellular locations CV file (The UniProt Consortium, 2015b) in a manner similar to the parsing of GO (Figure 5). 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 made the assumption 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:

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

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.

Creating category mappings from 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 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 (Supplemental Data 6 a, b).

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, a process summarized in Figure 1b. 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.

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 (McKinney, 2010) representation, and plotted the intersections using the pyUpSet python module (pyUpSet, 2016; Lex *et al*, 2014). 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.

Evaluating false mapping potential and possible true mapping pairs in GO

To determine how significant mapping issues are as a result 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 (pM_F) 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

$$PA_e = \{Ae_{child} + e_{child}\} - \{Ae_{par} + e_{par}\} \quad (2)$$

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 (PD_e) for every has_part edge (e) as

$$PD_e = \{De_{par} + e_{par}\} - \{De_{child} + e_{child}\} \quad (3)$$

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:

$$pM_{F,e} = \{(d, a) \mid d \in PD_e; a \in PA_e\} \quad (4)$$

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

$$pM_F = \left| \bigcup_{e=1}^{n} pM_{F,e} \right| \quad (5)$$

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

$$M_T = |\{n_1 anc \cap n_2 desc \mid n_1 \in G;\ n_2 \in G\}| \quad (6)$$

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.

Evaluating potential false mappings created by Map2Slim

Using the same method that we used to create mappings from M2S, we performed an all-against-all mapping for every term in GO, data-version: releases/2016-01-12. However, because M2S's custom term list option removes terms subsumed by other mappings, we were forced to 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 TORQUE 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 n a scoping paradigm, we defined the set of potentially false M2S mappings ($pM_{f,M2S}$) as

$$pM_{f,M2S} = \{M_{pair,M2S}\} - (\{M_{pair,M2S}\} \cap \{M_{pair,GOcats(scoping)}\}) \quad (7)$$

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

$$M2S_{error} = \frac{|pM_{f,M2S}|}{|\{M_{pair,GOcats(scoping)}\}|} \qquad (8)$$

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. Scripts for generating these results can be found in Supplemental Data 7.

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 (Binns *et al*, 2009) 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 IEA 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 8a). 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.

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 (Uhlen *et al*, 2015). Only gene product localizations 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 7). 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.

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 (Chris Mungall BDGP, 2013), 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 (GO Slim and Subset Guide). 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 (Supplemental Data 5).

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 (Flight *et al*, 2014)—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 (Huber & Gentleman, 2017). In this dataset, we ignored time point information and only considered data associated with the presence and absence of estrogen exposure. categoryCompare can consider GO ancestor terms for annotated terms in the experimental dataset when calculating enrichment. We therefore created two mapping dictionaries in Python where keys are each term in GO and values are 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 (Barrell *et* *al*, 2009). R scripts and Python scripts for generating the enrichment results can be found in Supplemental Data 8.

To compare FDR-adjusted (target FDR=0.05) 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.

Availability and Future Directions

The Python software package GOcats is an open-source project under the BSD-3 License and available from the GitHub repository https://github.com/MoseleyBioinformaticsLab/GOcats . Documentation can be found at http://gocats.readthedocs.io/en/latest/. All results are available on the FigShare repository https://figshare.com/s/cc1abe7e2e5c4ae09500 with the specific code used to generate these results found on the FigShare repository https://figshare.com/s/26b336a06946a9248e08.

We are actively developing the codebase and appreciate any contributions and feedback provided by the community. We are extending the API and adding additional capabilities to handle more advanced annotation enrichment analysis use-cases.

Acknowledgements

This work was supported in part by grants NSF 1252893 (Moseley), NIH 1U24DK097215-01A1 (Higashi, Fan, Lane, Moseley), and NIH UL1TR001998-01 (Kern). We thank Dr. Thilakam Murali for extensive feedback on the general scientific readability of the manuscript.

Author Contributions

E.W.H. worked on the design of GOcats, implemented GOcats, performed all analyses, interpreted results, and wrote over half of the manuscript. H.N.B.M. worked on the design of GOcats, troubleshooted various aspects of GOcats implementation, designed the analyses, interpreted results, and wrote a significant portion of the manuscript. R.M.F. implemented the command line interface for categoryCompare2 used for annotation enrichment analyses, helped design and interpret the annotation enrichment analyses, and revised the manuscript.

Conflict of Interest

The authors claim no conflict of interest.

References

Abeysinghe R, Hinderer EW, Moseley HNB & Cui L (2017) 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)*

Ashburner M, Ball C a & Blake J a (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25:** 25–29 Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/pmc3037419/

Assays and Annotation (2016) Available at: http://v15.proteinatlas.org/about/assays+annotation [Accessed September 1, 2017]

Barrell D, Dimmer E, Huntley RP, Binns D, O'Donovan C & Apweiler R (2009) The GOA database in 2009 - An integrated Gene Ontology Annotation resource. *Nucleic Acids Res.* **37:** 396–403

Binns D, Dimmer EC, Huntley RP, Barrell DG, O'Donovan C & Apweiler R (2009) QuickGO: a web-based tool for Gene Ontology searching. *Bioinformatics* **25:** 3045–3046 Available at: http://doi.wiley.com/10.1002/pmic.200800002

Bodenreider O (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* **32:** D267–D270 Available at: 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

Carlson M (2016) GO.db: A set of annotation maps describing the entire Gene Ontology. Available at: https://bioc.ism.ac.jp/packages/3.3/data/annotation/html/GO.db.html

Cesar J, Reis D, Santec CR, Tudor CRPH, Silveira M Da & Reynaud-delaître C (2013) Mapping Adaptation Actions for the Automatic Reconciliation of Dynamic

Ontologies. *Cikm***:** 599–608

Chris Mungall BDGP (2013) map2slim - maps gene associations to a ‘slim’ ontology. Available at: http://search.cpan.org/~cmungall/go-perl/scripts/map2slim

Cunningham F, Amode MR, Barrell D, Beal K, Billis K, Brent S, Carvalho-Silva D, Clapham P, Coates G, Fitzgerald S, Gil L, Giron CG, Gordon L, Hourlier T, Hunt SE, Janacek SH, Johnson N, Juettemann T, Kahari a. K, Keenan S, et al (2015) Ensembl 2015. *Nucleic Acids Res.* **43:** D662–D669 Available at: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1010

Data Quality Assurance and Scoring (2016) Available at: http://v15.proteinatlas.org/about/quality+scoring [Accessed September 1, 2017]

Flight RM, Harrison BJ, Mohammad F, Bunge MB, Moon LDF, Petruska JC & Rouchka EC (2014) Categorycompare, an analytical tool based on feature annotations. *Front. Genet.* **5:** 1–13

Gamma E, Helm R, Johnson R, Vlissides J & Booch G (1994) Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition Addison-Wesley Professional

Gene Ontology consortium (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.* **43:** D1049–D1056 Available at: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1179

Gene Ontology consortium (2017) Ontology Relations. Available at: http://www.geneontology.org/page/ontology-relations

GO Slim and Subset Guide Available at: http://geneontology.org/page/go-slim-and-subset-guide [Accessed November 22, 2016]

Götz S, García-Gómez JM, Terol J, Williams TD, Nagaraj SH, Nueda MJ, Robles M, Talón M, Dopazo J & Conesa A (2008) High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Res.* **36:** 3420–3435

Groß A, Pruski C & Rahm E (2016) Evolution of Biomedical Ontologies and Mappings: Overview of Recent Approaches. *Comput. Struct. Biotechnol. J.* **14:** 1–8 Available at: http://linkinghub.elsevier.com/retrieve/pii/S2001037016300319

Groß A, Dos Reis JC, Hartung M, Pruski C & Rahm E (2013) Semi-automatic adaptation of mappings between life science ontologies. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **7970 LNBI:** 90–104

Huber W & Gentleman R (2017) estrogen: Microarray dataset that can be used as example for 2x2 factorial designs.

Huttlin EL, Ting L, Bruckner RJ, Gebreab F, Gygi MP, Szpyt J, Tam S, Zarraga G, Colby G, Baltier K, Dong R, Guarani V, Vaites LP, Ordureau A, Rad R, Erickson BK, Wühr M, Chick J, Zhai B, Kolippakkam D, et al (2015) The BioPlex Network: A Systematic Exploration of the Human Interactome. *Cell* **162:** 425–440 Available at: http://www.sciencedirect.com/science/article/pii/S0092867415007680

Jiang JJ (1997) Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*

Jin H, Eun SY, Lee JSJH, Park SW, Lee JSJH, Chang KC & Kim HJ (2014) 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.* **16:** R77 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4406012&tool=pmcentrez&rendertype=abstract

Lex A, Gehlenborg N, Strobelt H & Vuillemot R (2014) UpSet□: Visualization of Intersecting Sets Supplementary Material. *IEEE Trans. Vis. Comput. Graph.* **20:** 1983–1992

Lin D (1989) An Information-Theoretic Definition of Similarity. In *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning* pp 296–304.

McKinney W (2010) Data Structures for Statistical Computing in Python. *Proc. 9th Python Sci. Conf.* **1697900:** 51–56 Available at: http://conference.scipy.org/proceedings/scipy2010/mckinney.html

Munoz-Torres M & Carbon S (2017) Get GO! retrieving GO data using AmiGO, QuickGO, API, files, and tools. *Methods Mol. Biol.* **1446:** 149–160

Na D, Son H & Gsponer J (2014) Categorizer: a tool to categorize genes into user-defined biological groups based on semantic similarity. *BMC Genomics* **15:** 1091 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4298957&tool=pmcentrez&rendertype=abstract

Noy N & Wallace E (2005) Simple part-whole relations in OWL Ontologies. *W3C.org*

Available at: https://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/

OWLTools (2015) Available at: https://github.com/owlcollab/owltools

Papatheodorou I, Oellrich A & Smedley D (2015) Linking gene expression to phenotypes via pathway information. *J. Biomed. Semantics* **6:** 17 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4404592&tool=pmcentrez&rendertype=abstract

pyUpSet (2016) Available at: https://github.com/ImSoErgodic/py-upset

Relations Ontology (2016) Available at: http://www.obofoundry.org/ontology/ro.html

Resnik P (1999) Semantic Similarity in a Taxonomy: An Information Based Measure and Its Application to Problems of Ambiguity in Natural Language. *J. Aritificial Intell. Res.* **11:** 95–130

van Rossum G & Drake F (2011) The Python Language Reference Manual Network Theory Ltd.

Schlicker A, Domingues FS, Rahnenführer J & Lengauer T (2006) A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics* **7:** 302 Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-33748335463&partnerID=tZOtx3y1

Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B & Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13:** 2498–504 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=403769&tool=pmcentrez

&rendertype=abstract

Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ, Leontis N, Rocca-Serra P, Ruttenberg A, Sansone S-A, Scheuermann RH, Shah N, Whetzel PL & Lewis S (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **25:** 1251–1255 Available at: http://www.nature.com/doifinder/10.1038/nbt1346

Storey VC (1993) Understanding semantic relationships. *VLDB J.* **2:** 455–488

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette M a, Paulovich A, Pomeroy SL, Golub TR, Lander ES & Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102:** 15545–50 Available at: http://www.ncbi.nlm.nih.gov/pubmed/16199517

Tang H, Klopfenstein D, Pedersen B, Flick P, Sato K, Ramirez F, Yunes J & Mungall C GOATOOLS: Tools for Gene Ontology. Available at: https://zenodo.org/record/31628 [Accessed October 3, 2016]

The UniProt Consortium (2015a) UniProt: a hub for protein information. *Nucleic Acids Res.* **43:** D204–D212 Available at: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku989

The UniProt Consortium (2015b) subcell.txt. Available at: http://www.uniprot.org/docs/subcell [Accessed May 27, 2015]

Uhlen M, Fagerberg L, Hallstrom BM, Lindskog C, Oksvold P, Mardinoglu a., Sivertsson a., Kampf C, Sjostedt E, Asplund a., Olsson I, Edlund K, Lundberg E, Navani S, Szigyarto C a.-K, Odeberg J, Djureinovic D, Takanen JO, Hober S, Alm T, et al (2015) Tissue-based map of the human proteome. *Science (80-.).* **347:** 1260419–1260419 Available at: http://www.sciencemag.org/content/347/6220/1260419

Veres D V., Gyurko DM, Thaler B, Szalay KZ, Fazekas D, Korcsmaros T & Csermely P (2015) ComPPI: a cellular compartment-specific database for protein-protein interaction network analysis. *Nucleic Acids Res.* **43:** D485–D493 Available at: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gku1007

Figures and Figure Legends

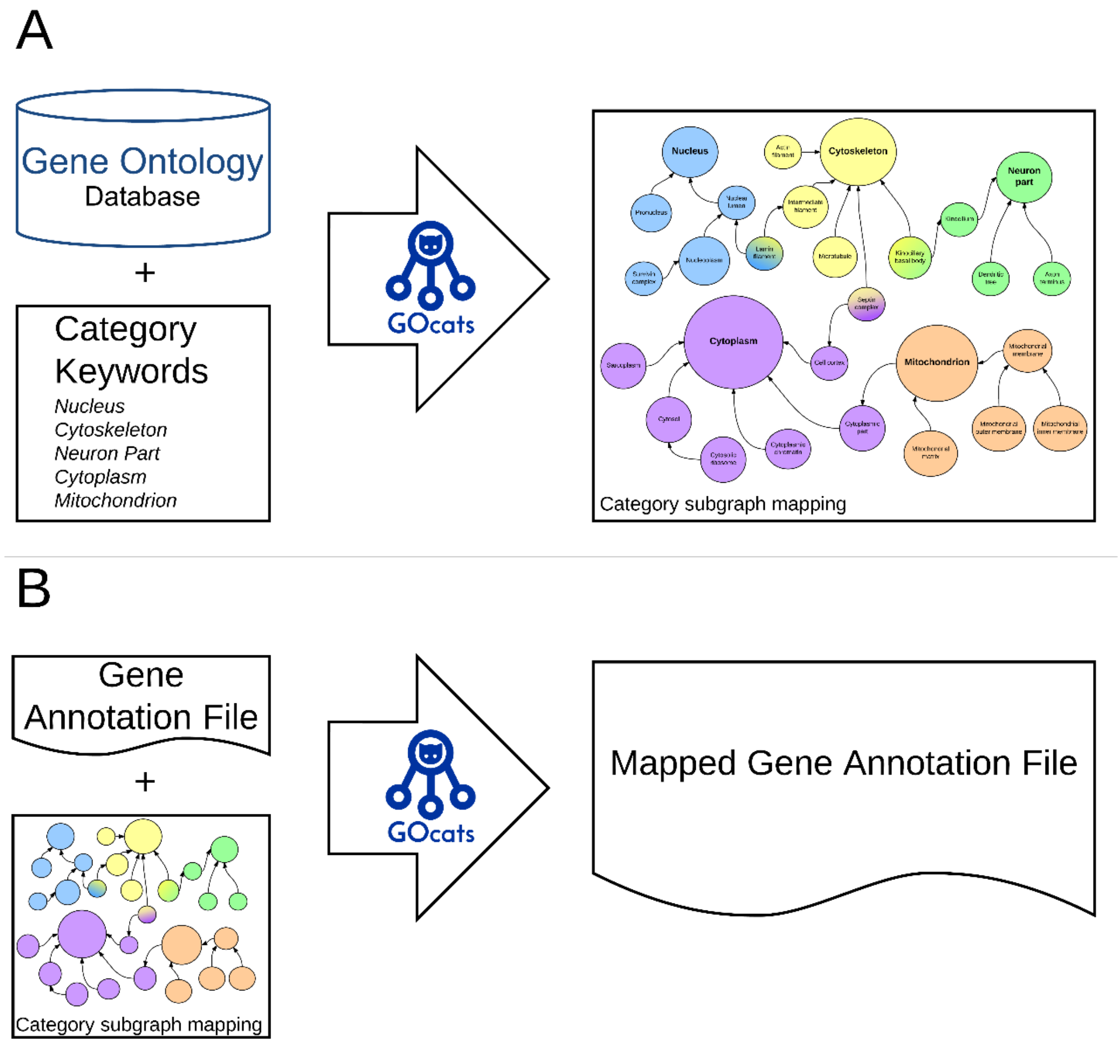


Figure 1 – GOcats data flow diagram for subgraph creation and GAF mapping.

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.

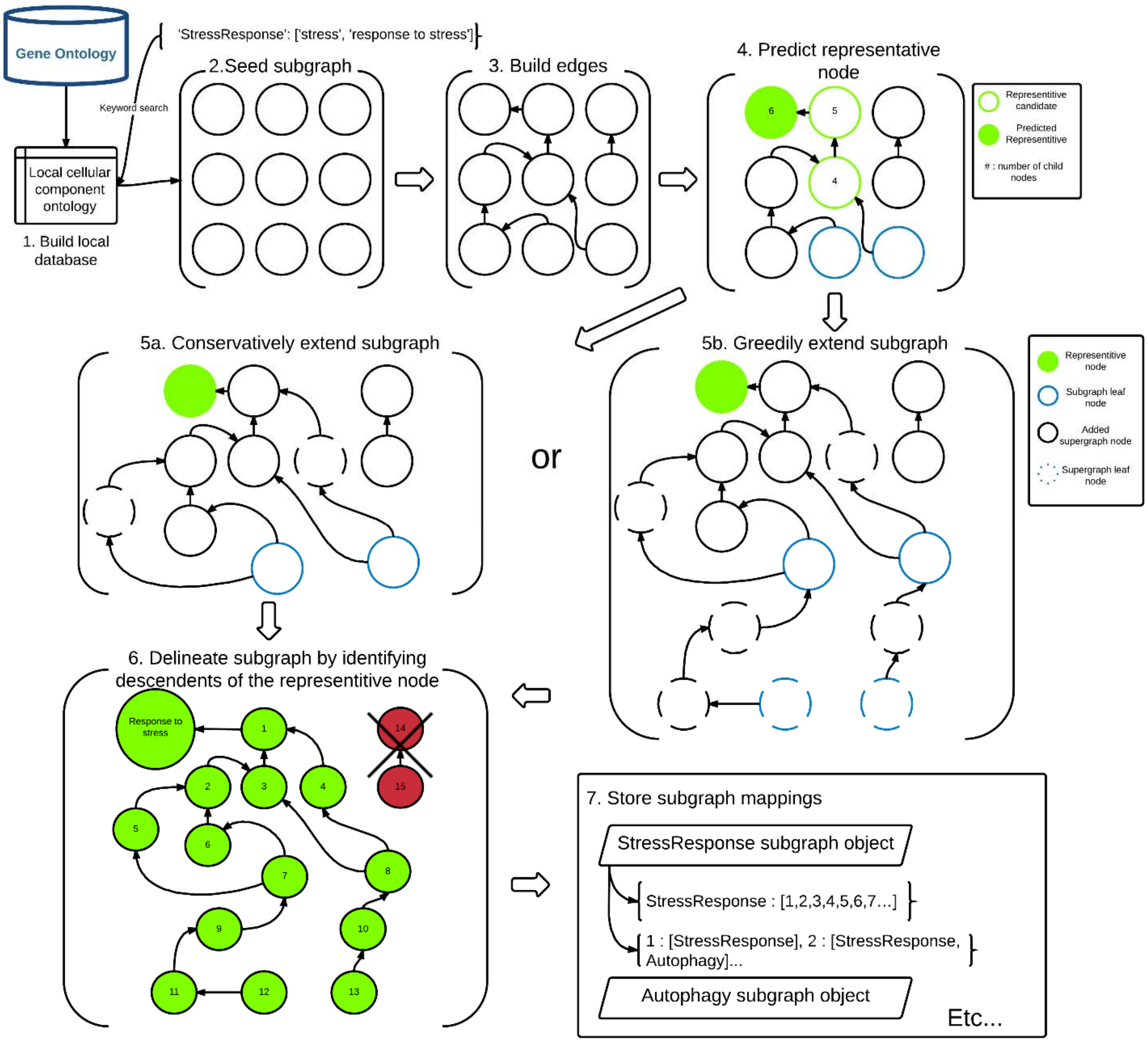

Gene Ontology
Keyword search
'StressResponse': ['stress', 'response to stress']
Local cellular component ontology
1. Build local database
2.Seed subgraph
3. Build edges
4. Predict representative node
6
5
4
Representitive candidate
Predicted Representitive
# : number of child nodes
5a. Conservatively extend subgraph
or
5b. Greedily extend subgraph
Representitive node
Subgraph leaf node
Added supergraph node
Supergraph leaf node
6. Delineate subgraph by identifying descendents of the representitive node
Response to stress
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
7. Store subgraph mappings
StressResponse subgraph object
StressResponse : [1,2,3,4,5,6,7…]
1 : [StressResponse], 2 : [StressResponse, Autophagy]...
Autophagy subgraph object
Etc...

Figure 2 – Flowchart of the GOcats' subgraph creation method. Individual steps occur in the designated numerical order, with conservative and greedy modes indicated by steps 5a and 5b, respectively.

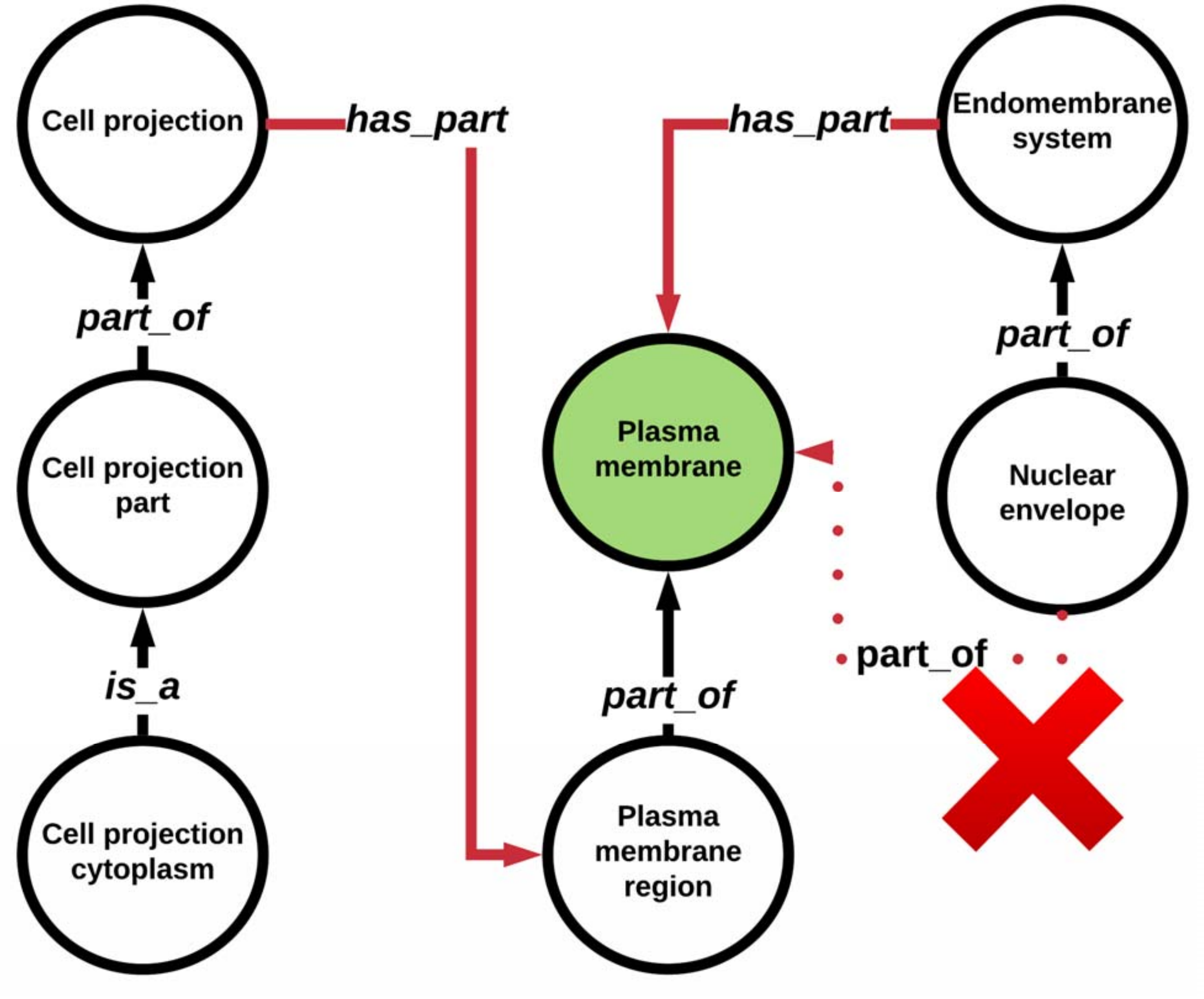


Figure 3 – The has_part relation creates paths with of varying semantic scoping which confuses mapping within GO. Some tools may create questionable GO term mappings, i.e. "nuclear envelope" to "plasma membrane," since the has_part relation edges point in from super-concepts to sub-concepts. GOCats avoids this by re-interpreting the has_part edges into part_of_some edges.

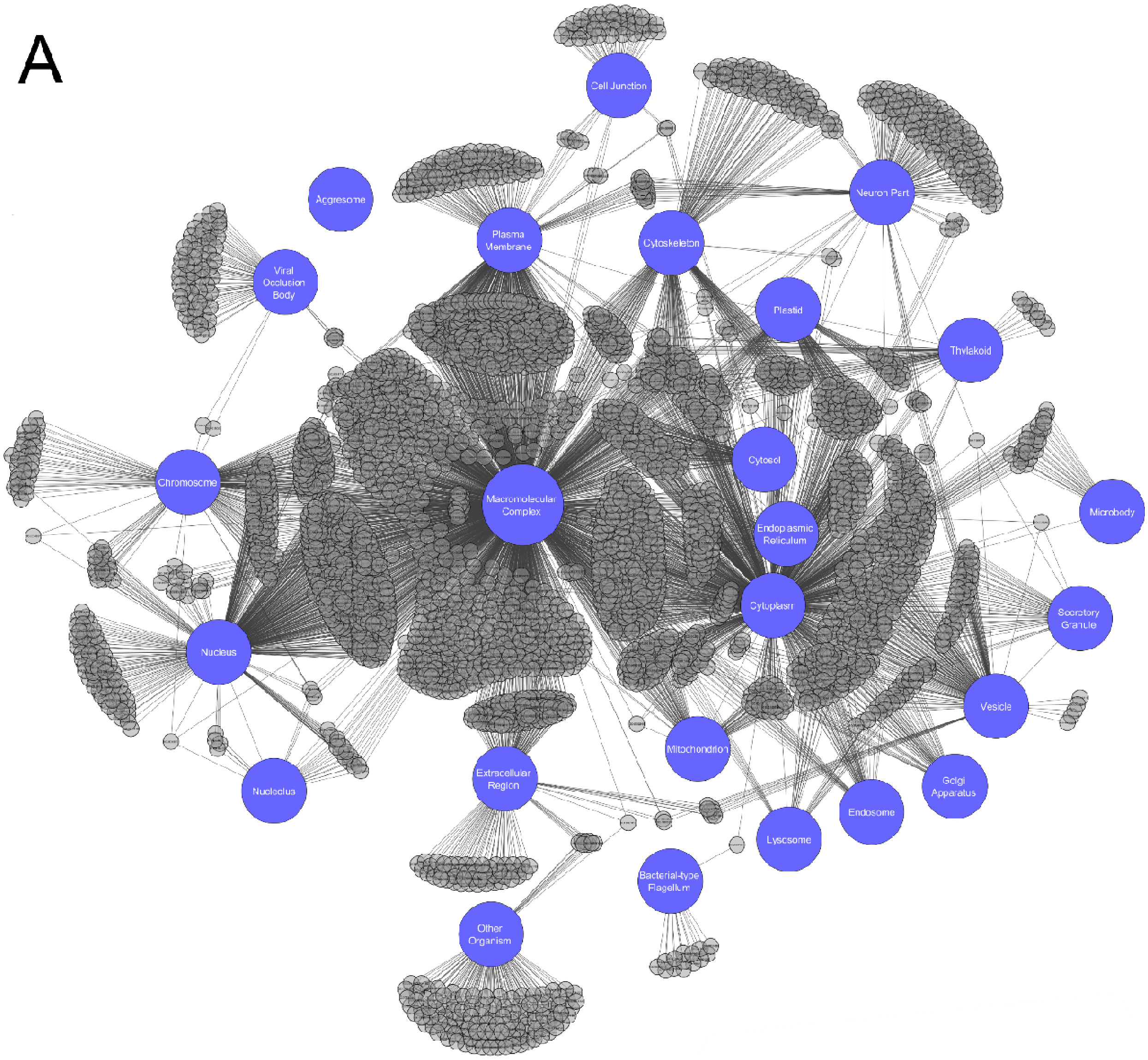
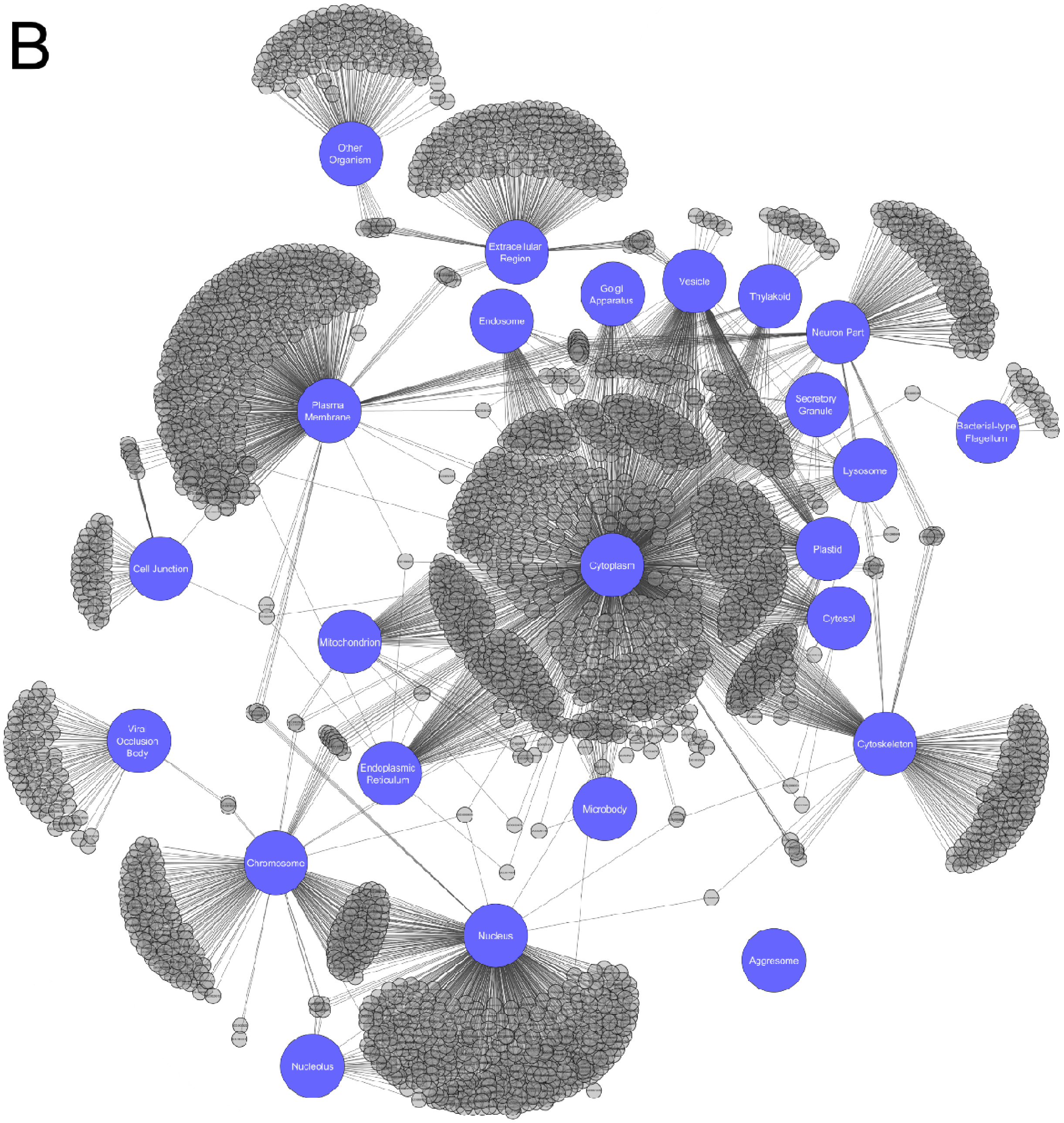
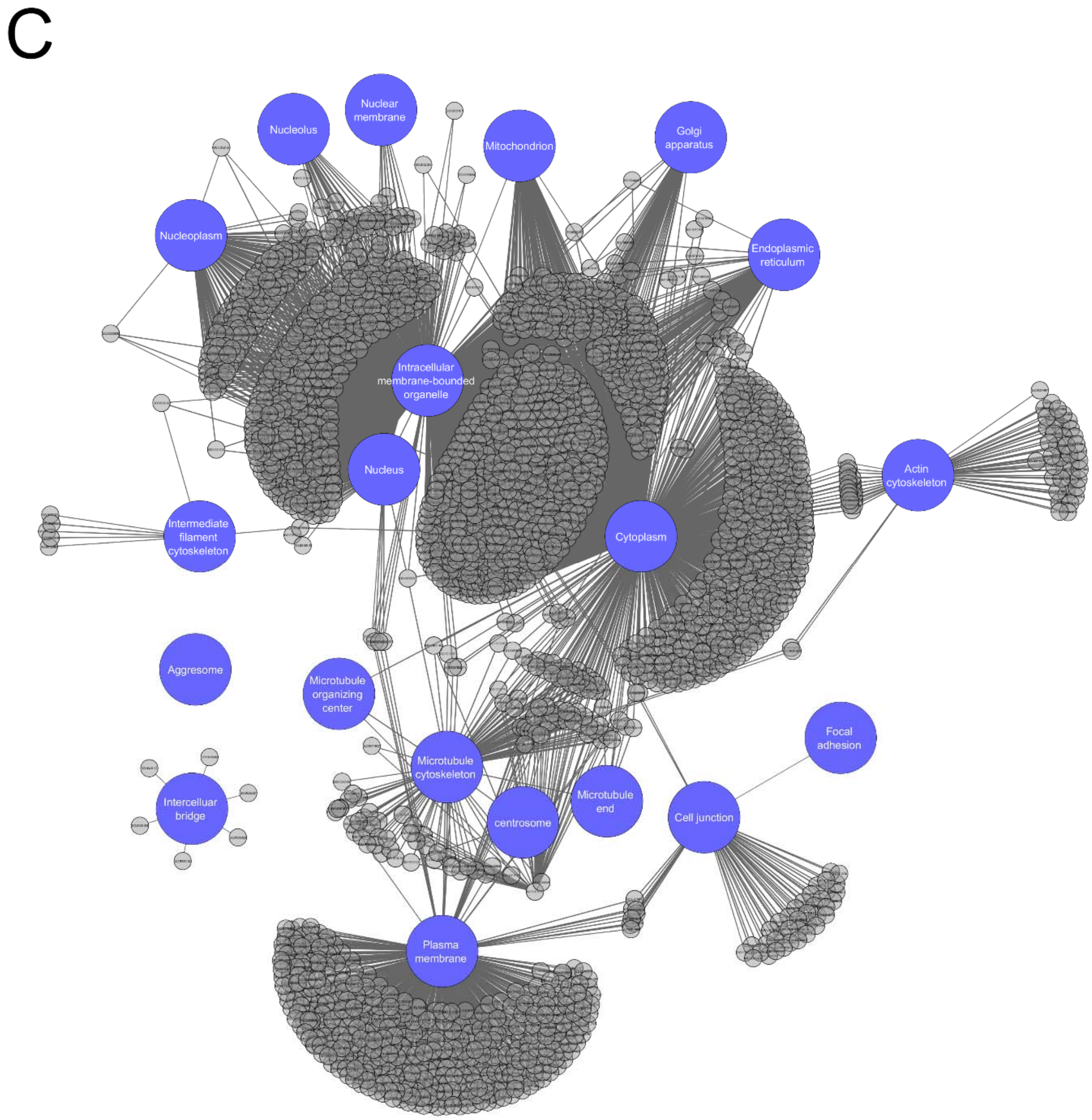


Figure 4 — Extracted GO subcellular localization subgraphs form a network with expected connectivity patterns. Blue nodes indicate category-representative nodes and grey nodes represent fine-grained GO terms that are part of their respective subgraphs. Edges connect fine-grained terms to their GOcats-assigned representative node(s). Images were created using Cytoscape 3.0 (Shannon *et al*, 2003).

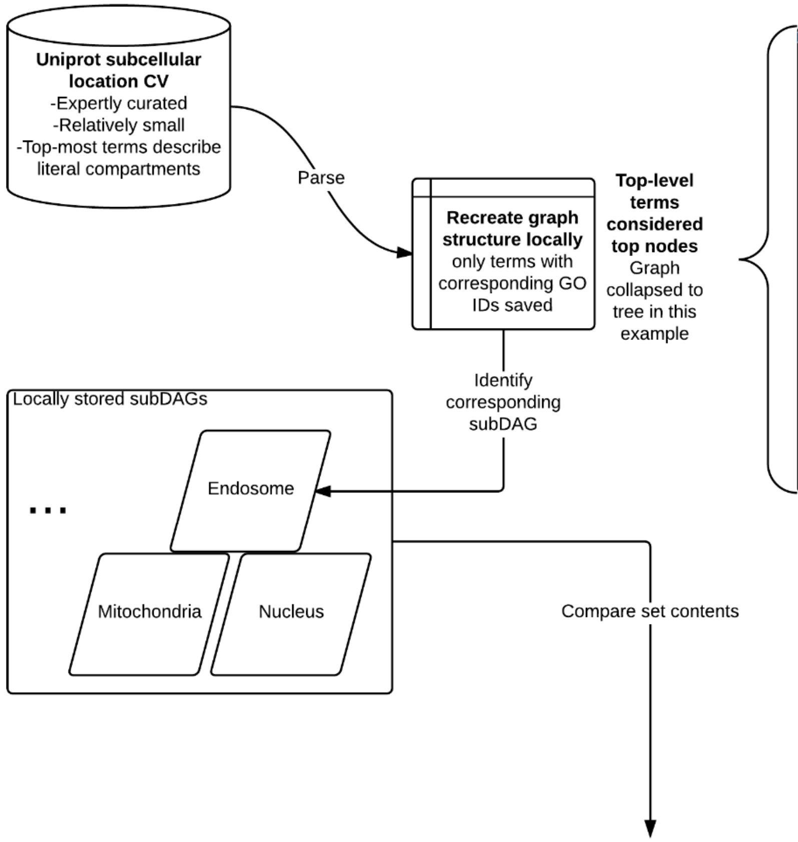
A) Network of 25 categories whose subgraphs account for 89% of the GO cellular component sub-ontology.

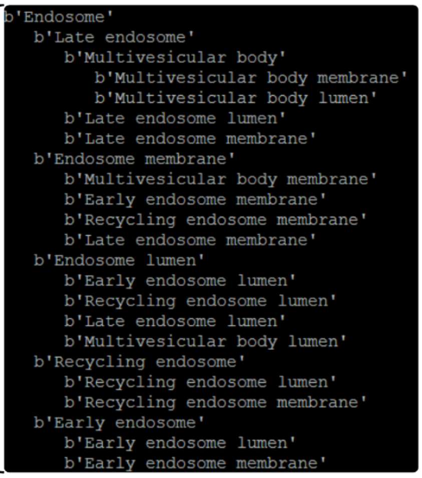


B) Network of 24 categories, with the "Macromolecular complex" subgraph left out for better visualization of the remaining categories.



C) Network of 20 categories used in the Human Protein Atlas subcellular localization immunohistochemistry raw data.


Uniprot subcellular location CV
-Expertly curated
-Relatively small
-Top-most terms describe literal compartments
Parse
Recreate graph structure locally
only terms with corresponding GO IDs saved
Top-level terms considered top nodes
Graph collapsed to tree in this example
Identify corresponding subDAG
Locally stored subDAGs
...
Endosome
Mitochondria
Nucleus
Compare set contents



```
b'Endosome'
   b'Late endosome'
      b'Multivesicular body'
         b'Multivesicular body membrane'
         b'Multivesicular body lumen'
      b'Late endosome lumen'
      b'Late endosome membrane'
   b'Endosome membrane'
      b'Multivesicular body membrane'
      b'Early endosome membrane'
      b'Recycling endosome membrane'
      b'Late endosome membrane'
   b'Endosome lumen'
      b'Early endosome lumen'
      b'Recycling endosome lumen'
      b'Late endosome lumen'
      b'Multivesicular body lumen'
   b'Recycling endosome'
      b'Recycling endosome lumen'
      b'Recycling endosome membrane'
   b'Early endosome'
      b'Early endosome lumen'
      b'Early endosome membrane'
```

*Redundancy is eliminated before finalizing the Uniprot subgraph

$$\begin{array}{c}\text{Inclusion}\\ \text{index}\end{array} = \frac{|\text{Uniprot subgraph}| \cap |\text{GO subDAG}|}{|\text{Uniprot subgraph}|}$$

Figure 5 – Flowchart of the UniProt subcellular location CV subgraph creation method and inclusion index equation.

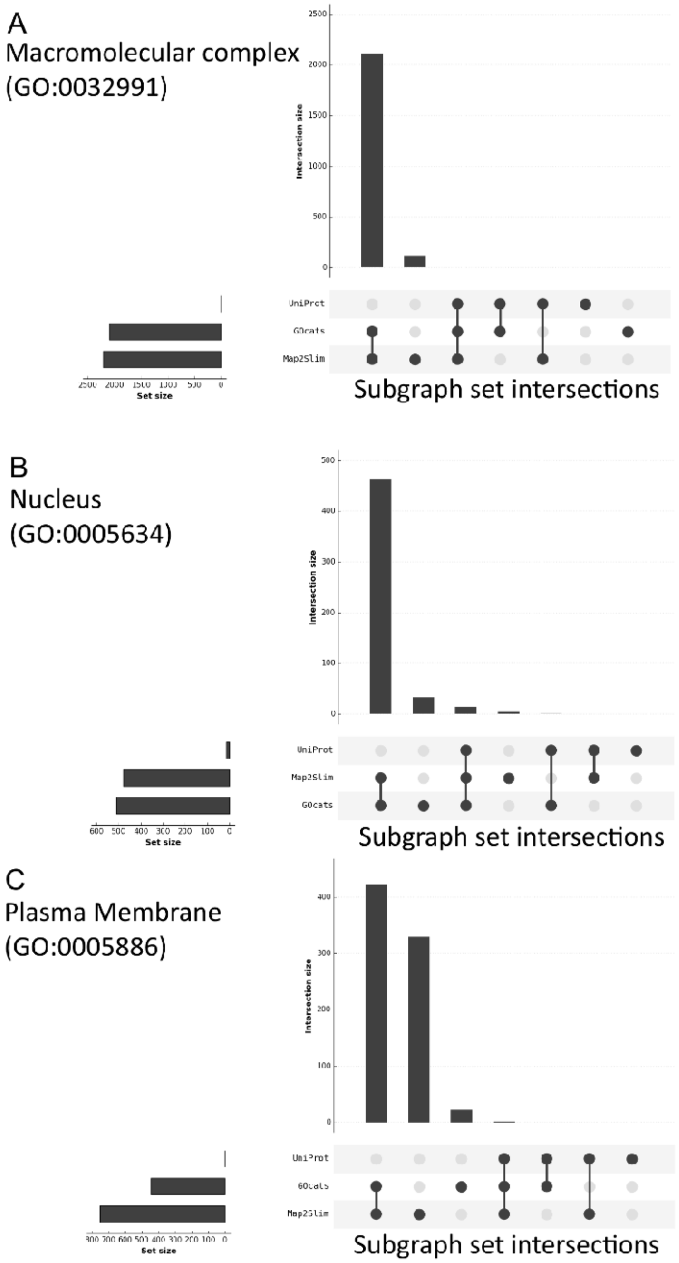


Figure 6 – Visualizing the degree of overlap between the category subgraphs created by GOcats, Map2Slim, and the UniProt CV. Plots were created using the Python package: PyUpset (pyUpSet, 2016) and represent the subgraphs created from mapping fine-grained terms in GO to the indicated general category using the indicated mapping method for: A) Macromolecular Complex; B) Nucleus; C) Plasma Membrane. Plots for the 22 additional categories can be found in Supplemental Data 1a-v.

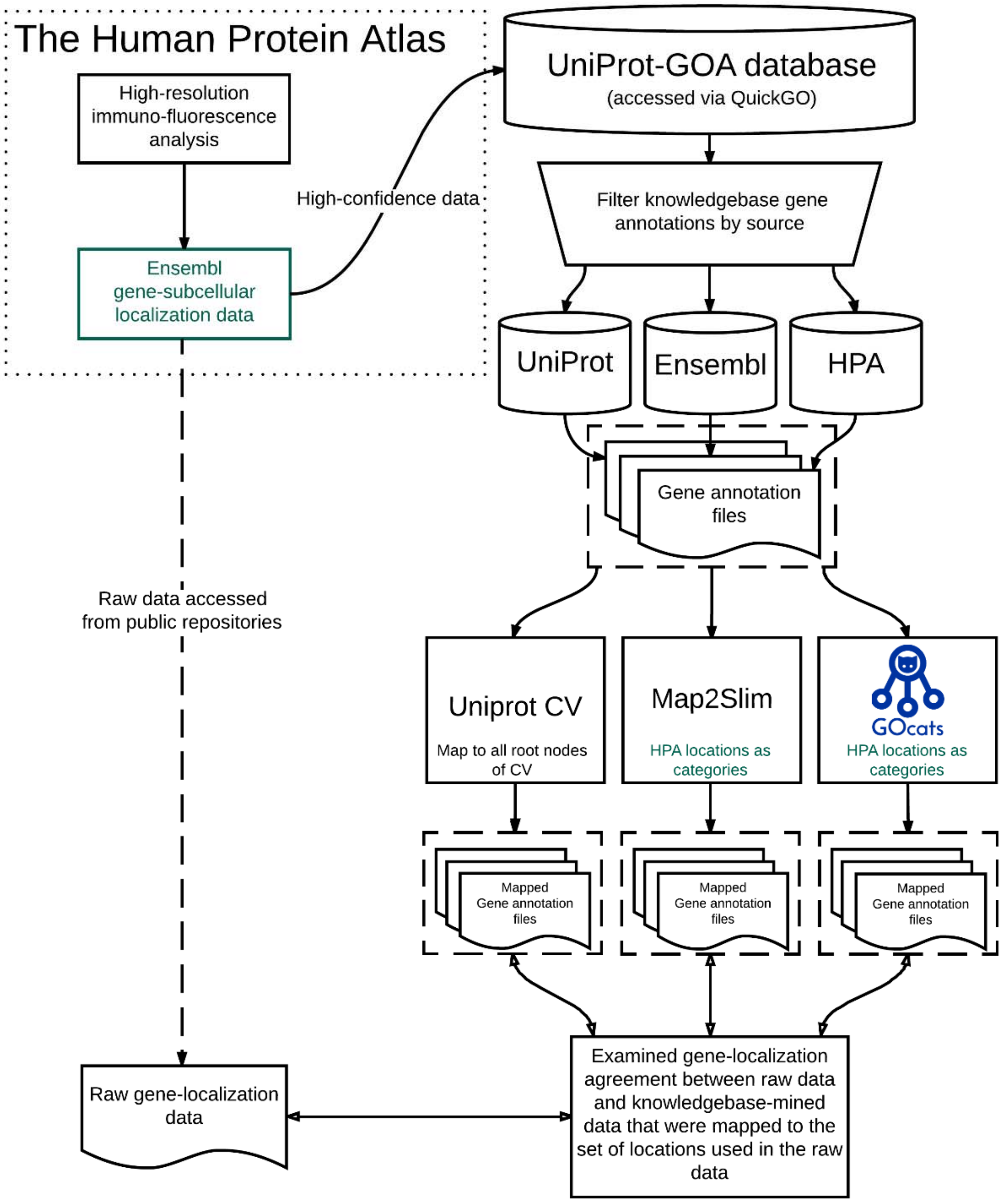

The Human Protein Atlas
High-resolution immuno-fluorescence analysis
Ensembl gene-subcellular localization data
High-confidence data
UniProt-GOA database
(accessed via QuickGO)
Filter knowledgebase gene annotations by source
UniProt
Ensembl
HPA
Gene annotation files
Raw data accessed from public repositories
Uniprot CV
Map to all root nodes of CV
Map2Slim
HPA locations as categories
GOcats
HPA locations as categories
Mapped Gene annotation files
Mapped Gene annotation files
Mapped Gene annotation files
Raw gene-localization data
Examined gene-localization agreement between raw data and knowledgebase-mined data that were mapped to the set of locations used in the raw data

Figure 7 – Methods overview of knowledgebase gene annotation mapping and comparison to Human Protein Database subcellular localization raw data.

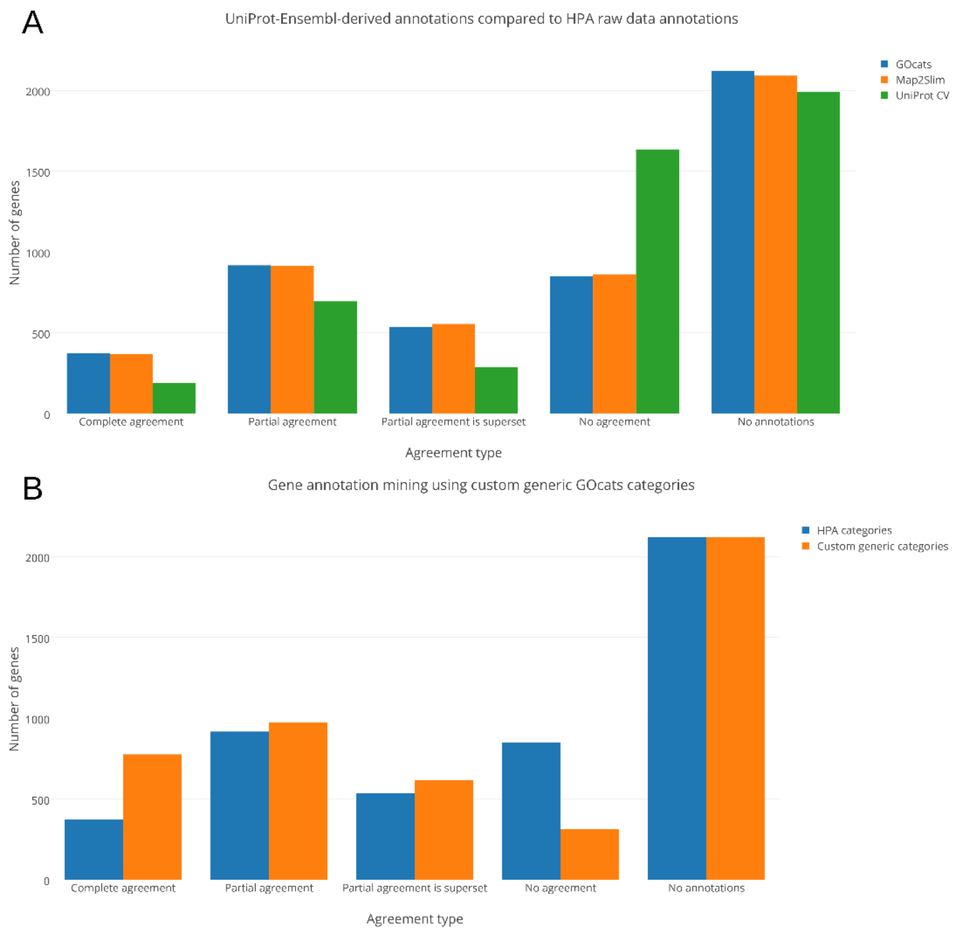


Figure 8 – Comparison of UniProt-Ensembl knowledgebase annotation data mining extraction performance by GOcats, Map2Slim, and UniProt CV. “Complete agreement” refers to genes where all subcellular locations derived from the knowledgebase and the HPA dataset matched, “partial agreement” refers to genes 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), "no agreement" refers to genes with no subcellular locations in common, and “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 8.

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.

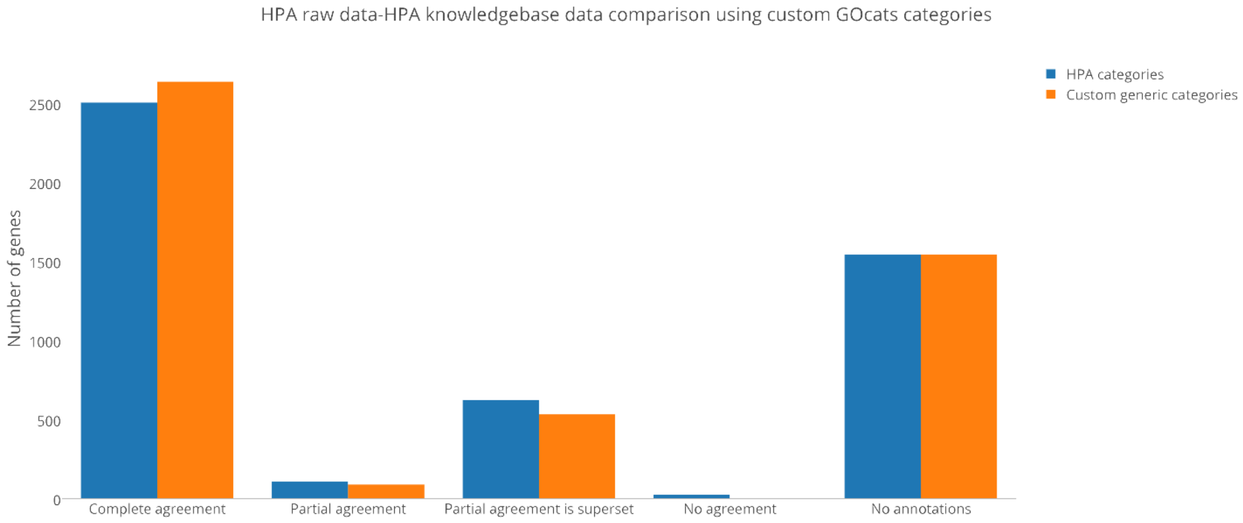


Figure 9 – Comparison of HPA knowledgebase derived annotations to HPA experimental data. Number of genes in the given agreement type when comparing gene product annotations assigned by HPA in the EMBL-EBI knowledgebase to those in The Human Protein Atlas' raw experimental data. "Complete agreement" refers 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, and "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 8.

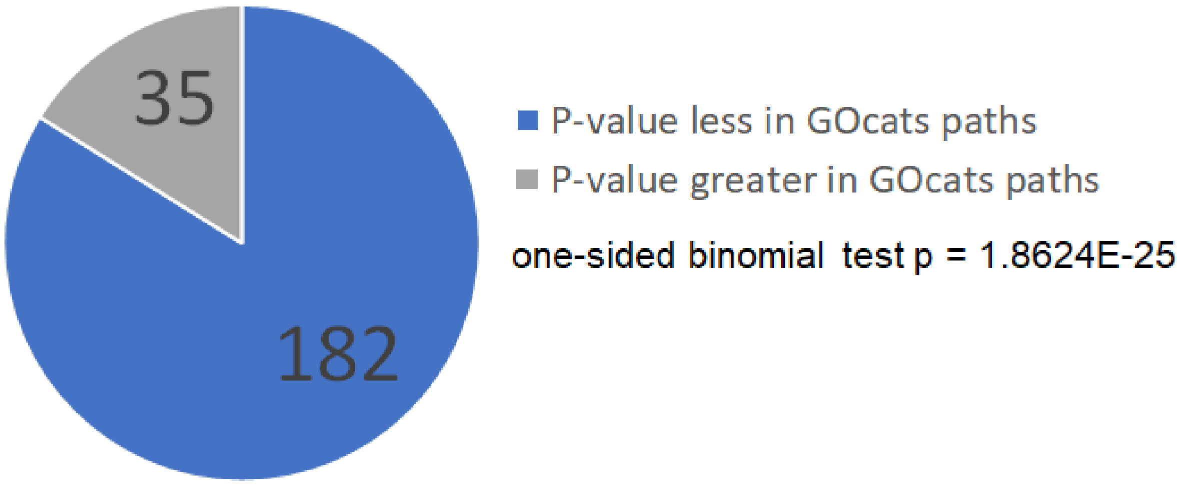


Figure 10 – Comparing p-values of significantly-enriched annotations using GOcats paths vs excluding has_part edges. The majority of significantly-enriched GO terms had an improved p-value when GOcats re-evaluated has_part edges for the enrichment of the cancer data set in this investigation.

Tables and Table Legends

Table 1 – Prevalence of relations in the Gene Ontology and suggested semantic correspondence classes to reduce ambiguity.

Relationship	Prevalence in GO All sub-ontologies	Prevalence in GO Cellular Component	Prevalence in GO Biological Process	Prevalence in GO Molecular Function	Correspondence Class	Correspondence members
is_a	72455	5591	54689	12175	Scoping	hyponym "is_a"

					(hyponymy)	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	Spaciotemporal (process-process)	process "ends_during" process
occurs_in	181	0	180	1	Spaciotemporal (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

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

Table 2 – 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 retuculum (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	1317	969	184	2102

		(GO:0032991)				
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
Vessicle	vesicle, vesicles	vesicle (GO:0031982)	198	90	85	203
Viral	virion, virus, viral	viral occulsion body (GO:0039679)	93	1	26	68

🟩	Expected representative
⬜	Unexpected representative

Table 3 – 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	GO:00057	0.818182	0.0873786	101	11

Reticulum	83				
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 4 – Agreement summary between corresponding GOcats and Map2Slim subgraphs.

Location Category	Term ID	Inclusion Index‡	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:000	0.984375	0.88317	211	192	40

	5694		8			
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 5 – Prevalence of potential has_part relation mapping errors in GO.

Sub-Ontology	Estimated Potential False Mappings (epM$_F$)	True Mappings (M$_T$)	M$_T$ $\cap$ epM$_F$	Potential False Mappings pM$_F$ = epM$_F$ - (M$_T$ $\cap$ epM$_F$)	True Mappings without HP ($_{IA_PO}$M$_T$)*	Lost Mappings (M$_T$ - $_{IA_PO}$M$_T$)*
Cellular component	30036	56025	6396	23640	49679	6346
Molecular function	10074	62436	1746	8328	56194	6242
Biological process	93092	555543	327	89815	527869	27674

			7			

* IA_PO refers to a graph created with only is_a and part_of relationship edges.

Table 6 – Summary of GO term mapping errors resulting from misevaluation of relations with respect to semantic scoping.

Ontology	**Map2Slim Mappings ($M_{pair,M2S\_ont}$)***	**GOcats Scoping Mappings ($M_{pair,Gocats\_ont}$)***	**Potentially false Map2Slim Mappings $pM_{F,M2S}$ = $M_{pair,M2S}$ - ($M_{pair,M2S} \cap M_{pair,Gocats\_all}$)***	**Map2Slim Correct Mappings $M_{T,M2S}$ = $M_{pair,M2S} \cap M_{pair,Gocats\_all}$***	**Possible Map2Slim Percent Error $pM_{F,M2S}$ / $M_{pair,M2S\_ont}$**
All GO	1036141	820467	325180	710961	0.313837595
Cellular Component	71835	56025	22059	49776	0.307078722

Molecular function	86163	62436	29955	56208	0.347655026
Biological process	878143	555543	273166	604977	0.311072342

* 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 7 – 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	117	22	77	62

		(GO:0015629)				
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 retuculum (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[‡])	intrecellular 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, mitochonrdial, mitochondrion	mitochondrion (GO:0005739)	134	2	44	92
Nuclear membrane	nuclear memmbrane	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

‡ HPA conservatievly annotates "vessicles" as intracellular membrane-bounded organelle

🟩	Expected representitive
⬜	Unexpected representitive

Table 8 – Generic location categories used to resolve potential scoping inconsistencies in HPA raw data.

HPA annotation category	GOcats-customized general HPA category
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 9 – Summary of gene location category agreement between manually-curated HPA raw data and GOcats/Map2Slim categorized HPA-derived knowledgebase annotations.

	Agreement				
Location	Complete	Partial	Superset[‡]	None	Not in Knowledgebase
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

‡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.

Supporting Information Legends

S1 – Visualizing the degree of overlap between the category subgraphs created by GOcats, Map2Slim, and the UniProt CV (additional categories). Plots were created using the Python package: PyUpset (pyUpSet, 2016) and represent the subgraphs created from mapping fine-grained terms in GO to the indicated general category using the indicated mapping method.

S2 – List of GO terms mapped by Map2Slim to the term "plasma membrane" (GO:0005886) that were not mapped to this location by GOcats.

S3 – Comparing adjusted p-values between omitted has_part and GOcats' part_of_some edges. Only terms with adjusted p-values < 0.01 in the omitted has_part version were included for clarity. The full list of GO terms from the enrichment analysis can be found in the results directory of S8.

S4 – Uniquely enriched terms between GOcats paths and traditional paths.

S5 – Python and shell scripts used to verify that the output of the Java and Perl versions of Map2Slim are identical given the same dataset and GO term list.

S6 – Map2Slim data flow diagram and our alternative for producing a standalone GO term mapping.

A) The Java version of Map2Slim is able to take a list of GO terms (or a GO slim file), a gene annotation file (GAF) and a locally-saved GO database file, to create a GAF with fine-grained terms mapped to the set of GO terms in the GO slim file or the provided term list.

B) To create a standalone mapping file, we generated a custom GAF where each GO term in the ontology was represented once and each gene name was renamed to match each GO term ID. The resulting mapped GAF can then be parsed to create a standalone mapping of all terms in GO to the set of slim terms.

S7 – Python, Perl, and shell scripts used to generate every GO term-to-GO term mapping possible with Map2Slim.

S8 – Scripts for performing annotation enrichment of breast cancer data.