GRADUATE THESIS/DISSERTATION APPROVAL FORM AND SIGNATURE PAGE

Instructions: This form must be completed by all master's and doctoral students with a thesis or dissertation requirement. Please type or print clearly as this form MUST be included as page 1 of your thesis or dissertation via electronic submission to ProQuest. All theses and dissertations must be formatted according to the University and department/program requirements. **Reminder:** It is the responsibility of the student to submit any/all edits requested by the Examining Committee to the Faculty Mentor or Supervising Professor for final approval and signature via the Graduate Program Completion Form.

Type: ☐ Master's Thesis ☑ PhD/Doctoral Thesis or Dissertation

Thesis or Dissertation Title: Annotation, Significance, and Their Conservation in Animal Models (mouse, fruit fly)

Author's Name: D. V. Klopfenstein

Month and Year: March 2021

The signatures below certify that this thesis / dissertation (circle one) is complete and approved by the Examining Committee.

Committee Chairperson's Name: Dr. Andres Kriete

Title: Professor, Associate Dean for Academic Affairs

Department: Biomedical Engineering

Institution (*if other than Drexel University*):

Signature:

Committee Member's Name: Dr. Ming Xiao

Title: Associate Professor

Department: Biomedical Engineering

Institution (*if other than Drexel University*):

Signature: Ming Xiao Digitally signed by Ming Xiao DN: cn=Ming Xiao, o=Drexel University, ou=School of Biomedical Engineering, email=mx44@drexel.edu, c=US Date: 2021.04.01 11:30:41 -04'00'

Committee Member's Name: Dr. Felice Elefant

Title: Professor, Director of Biology Graduate Program

Department: Biology

Institution (*if other than Drexel University*):

Signature: Felice Elefant

Committee Member's Name: Dr. Will Dampier

Title: Assistant Professor

Department: Biomedical Engineering

Institution (*if other than Drexel University*):

Signature:

Committee Member's Name: Dr. Ahmet Sacan

Title: Assistant Professor

Department: Biomedical Engineering

Institution (*if other than Drexel University*):

Signature:

Committee Member's Name: Dr. Christophe Dessimoz

Title: SNSF Professor

Department: Center for Integrative Genomics

Institution (*if other than Drexel University*): Universite de Lausanne, Switzerland

Signature:

Protein-Coding Hotspots in the Human Genome: Annotation, Significance, and Their Conservation in Animal Models (mouse and fruit fly)

A Thesis

Submitted to the Faculty

of

Drexel University

by

D. V. Klopfenstein

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

March 2021

Dedications

To my mom and dad, MaryAnn and Alan, with all my Love.

And to Brad, with all my heart.

Acknowledgments

Nestled close between my parents, happy and warm in the dark room with a small black-and-white television sitting not too far away, Mom and Dad were brimming with excitement. On the small screen, together we intently watched that most famous first step onto the moon. Neil Armstrong spoke: “One small step for man, one giant leap for mankind.”

Watching the moon landing snuggled with my Mom and Dad was my first memory in life.

Hearing “man,” I asked about woman. Mom and Dad assured me that “man” meant everybody. Phew, thank goodness. My mom and dad had moved to Huntsville, Alabama, where I was born, to do their part to support the moon shot. They were employees of IBM, a contractor to NASA, the National Aeronautics and Space Administration. Dad worked on keeping the equipment cool in a ring of the Saturn rocket; Mom worked on the software in the rocket that communicated with Earth.

Ones and zeros

As a child, at night I enjoyed removing the back panel from my radio, an old one, to feel the warmth and watch the glow of the vacuum tubes. A vacuum tube is so hot that you dare not touch it while it glows, lest you burn yourself.

Mom told me that if you made errors when writing your computer program, resulting in it failing to run, it was “due to bugs.” Dad told me that the old computers owned by big companies were made from vacuum tubes, just like the ones in my radio. But now, computers like the one that my Mom

bought our family (one of the first Apple computers), are made from transistors, which are microscopic and much cooler.

Dad said that if the programs did not work when run on the old vacuum tube computers, it was often due to actual insects (bugs). The warmth of the tubes would lure them near, but then the bugs would get too close, touch the glass tube and be fried dead. Often, this would cause the tube to fail and go dark and cold, and then the programs would not work. That is why the errors in a program are called bugs.

Dad told me that vacuum tubes are used as switches and so are transistors. He toggled the light switch in my room. "See? On and off. One and zero. It is digital, as opposed to analog, which is like a dimmer switch." He told me that a computer is made of thousands and thousands of switches.

Numbers need to be expressed in binary notation, ones and zeros, to fit into the switches that comprise a computer, continued Dad. Hex notation would be introduced to me when using Mom's work calculator and seeing letters as well as numbers in the display.

Computer camp

In computer camp during the summer before tenth grade, we learned a whole bunch of languages. My favorite was APL, because a program written in APL looked like gobbledygook, like a secret code.

Mom and Dad

I thank my Mom, MaryAnn Diesburg, and Dad, Alan Klopfenstein, for instilling in me a deep passion for technology by doing things such as ensuring that the family always had a computer, back in the days when almost no households (including IBM families) had one, and for sending me to computer camp.

Mom instilled the fight and fire in me necessary to overcome the challenges. At IBM, she was an ace saleswoman who would often bring sales (and commissions) far higher than her peers by selling to companies in Silicon Valley and other tech hot spots. Some of her marketing magic hopefully rubbed off on to me. Mom always picks up the phone when I call, even if I am not at my best and might be feeling quite stressed about the PhD. Thank God for those calls; what a lifeline. I hope I did not worry her too much.

Dad provided a sense of equanimity to slog through all the labor required to master a subject. My love of fitness came from our hikes together through the woods and up and down mountains. This too would serve me well. I grieve that my Dad has passed and is unable to be with Mom and me at the completion of my education at Drexel. I will forever miss our long conversations about tech and life in general. He would have beamed with pride to see all the progress leading to the successful publication of my peer-reviewed papers.

With this background and support, of course tech was to be my future. What else made sense? My Dad was a mechanical engineer, and that field called to me. But the folks at Rensselaer Polytechnic Institute (RPI) in Troy,

New York, said that majoring in electrical engineering was harder, so it seemed more challenging to do that, and I do not shirk from a challenge.

The decision turned out to be the right choice. After graduating and entering the workforce, it was great fun helping to "build a better mousetrap" by contributing to constructing products that worked well right off the shelf by creating verification environments, behavioral models, scoreboards, and a pleasant user interface aimed at hardware design engineers. My peers, almost always all men, liked that I made them look good, enabling them to catch flaws in their designs before anyone noticed.

The switch

While working in electrical engineering, I long contemplated switching fields to an engineering field in biology or medicine. It would require a return to university. It was only starting from about 2005 that improvements in computer languages, computer hardware, and databases relying on the completion of the human genome in 2003 would make bioinformatics in biomedical engineering a sensible career choice for my skills and interests.

My growing interest in health knowledge led me to volunteer as an Emergency Medical Technician (EMT) on my town's ambulance for over ten years starting just before 2000. If you called 911 on a Tuesday evening, I was in the crew that arrived to treat and transport you to the hospital if injured or ill or needing help.

Carrot and stick: Besides the lure of the carrot of dreaming to help groups of scientists make biological discoveries, there was also the stick from being

the only woman in an electrical engineering group. My peers in my working group were all men.

So many tech women were being laid off all over the company that upper management talked about converting one of the two women's restrooms into a men's restroom. Knowing my skills and work environment at that time, a former manager at a different company offered me a different job in electrical engineering, if I had had enough.

But the real tipping point in choosing to leave electrical engineering altogether was SystemVerilog (SV), a horrid computer language that upper management was shoving down upon us. Computer Aided Engineering (CAE) companies had been pushing this awful half-baked language for over ten years without it catching on. It was supposed to replace successful verification environments written in either of the hardware languages, VHDL or verilog, plus C++, a beautiful language used by millions and millions of people in a wide variety of fields. Young people today entering the field of application specific integrated circuit (ASIC) verification don't even know that there were successful environments before SystemVerilog came on to the scene.

SV is grotesquely patched, Frankenstein-style, from the essence of VHDL and verilog with promises of C-ishness. SV is non-standard from company to company, used only in one tiny sub-field, poorly documented, slow to compile, slow to run, has virtually no support, little community, and knowing it provides no job prospects beyond the diminishing world of hardware verification. After ten-plus years of development and fixes, CAE companies finally had an almost passable version of SV and were pushing it hard.

No thanks. While seriously considering leaving electrical engineering anyway, this seemed the ideal time.

Dr. Will Dampier

Thank you to my incredible, brilliant, tech-savvy advisor. He knows what the command-line is and how to use it. It is so gratifying to work closely with someone who is simpatico regarding a fundamental tool suite that is useful in getting work done.

One of our first conversations when he became my official advisor was regarding my first paper, which was under development. He was excited and fully supportive. The project and paper were massive, all-consuming efforts. Needing a solid paper, I threw everything I had into the task. It paid off. My cool, young, amazing, hip professor has over 130 papers. And my first paper rose to number 2 out of all ~130 if his papers are sorted by the number of papers that cited them (citation count).

My first first-author paper ended up performing well, but its birth and development were utter pain. My messy and rambling early publication drafts would have never resulted in clear, informative peer-reviewed research publications without the support of both Dr. Dampier and Carol Berman. Dr. Dampier has remarkable instincts regarding how to focus the text so that my papers are pointed and clear, emphasizing only the most important material.

Dr. Dampier's advice regarding figures in my papers was pure magic. His recommendations resulted in new figures that captured the content of my text

comprehensively and accurately, making the paper friendlier and more inviting for the reader.

Dr. Andres Kriete

How do I turn a chemical equation into a mathematical equation? I was stumped until I experienced Dr. Kriete's biological modeling class. Dr. Kriete's expert instruction is the sole reason that I correctly answered one of the questions on the written portion of the PhD qualifying exam. The book Dr. Kriete required for his class has a special spot on my bookshelf and is now quite dog-eared and worn.

Dr. Ming Xiao

My introduction to Dr. Xiao began with him describing his research, which was about his cool optical mapping of long, individual DNA molecules into nanochannel arrays. He played movies of the DNA strands, which were jumbled and collecting at the funnel by the entrance of the long slender channels. These channels were arranged to be parallel, like swimming pool lanes. Like magic, the DNA strands would untangle and then a single strand would enter each channel where it could be mapped. The DNA movies and Dr. Xiao's explanations felt like story time.

Dr. Xiao is particularly interested in how changes in telomeres, found on the ends of our chromosomes, might by associated with aging, age-related diseases, chemical exposure, DNA damage, and tumor development. His

research has been crucial to mapping repetitive sequences in our DNA, which are notoriously difficult to sequence.

I created some figures for him and worked as a teaching assistant (TA) in his biostatistics class. It was a great class with lots of interesting problem sets, including Dr. Sacan's comprehensive microarray set. But the best part was interacting with students during office hours. The problem sets were difficult, but Dr. Xiao cared about his students and worked directly with them and the TAs so that we could best help the students comprehend the material. It was a wonderful and unforgettable experience.

Dr. Ahmet Sacan

Taking classes part-time while working full-time during the master's portion of the curriculum was exhausting. There were periods when time just ran out. But Dr. Sacan never let us off the hook. He would smile and say, "Just go solve the problem." Sigh. So weary. But I would do it, finding the result so gratifying, boosted from the adrenaline rush of fully absorbing something new.

My favorite Sacan problem set was one encountered while working as a teaching assistant for Dr. Xiao, who used Dr. Sacan's problem set in his class. It was a comprehensive micro-array data set with many associated questions. Working through it with the students in the class was immensely satisfying and educational for both me and the students. That was my favorite problem set of all, and there were some pretty good ones in that class.

Dr. Felice Elefant

To get a better handle on biology, I took one of Dr. Elefant's classes from Drexel's biology department. It was tremendously educational, but the best experience was the final student project and presentation.

My assigned project was to read an important research paper about schizophrenia, interpret it, and then present my understanding of the new insights from the paper. It was so informative to do the research and presentation and would serve me well later, when patterns for Schizophrenia showed up in runs of consecutive genes in my PhD data.

Dr. Christophe Dessimoz

Dr. Dampier suggested that I invite one of my co-authors onto my committee. There were thirteen other authors on my first first-author paper, but the choice was clear. While I was working on the GOA TOOLs paper, Dr. Tang asked my opinion regarding inviting Dr. Dessimoz on to the paper for his contributions in gene ontology semantic similarity. Of course, I agreed.

Upon inviting Dr. Dessimoz, he responded in no uncertain terms that if his student, Alex Warwich Vesztrocy, who wrote a tutorial for a book chapter about working with gene ontology using GOA TOOLs in Dessimoz's book, was not invited too, then "no way," he would not do it.

Wow. A man who will fight tooth and nail on behalf of his students. That is the guy I want in my corner. So it was a no-brainer to choose him from all other co-authors to join the committee.

Dr. Haibao Tang

What a fantastic experience it has been working on the open-source project, GOA TOOLs, a library for gene ontology analyses. Thank you so much to its brilliant project owner, Dr. Tang, for inviting me to be a collaborator and principal lead after assessing my early contributions to GOA TOOLs, which were necessary for my own research.

While I wrote my first peer-reviewed paper, he challenged me to create a set of stochastic simulations of gene ontology enrichment analyses (GOEA) for the publication. Architecting and implementing the extensive set of simulations culminated in creating and answering questions regarding the effect of various GOEA parameters, which was previously unknown and caused the results to vary dramatically.

Dr. Tang's invitation into GOA TOOLs expanded my horizons from the insular world of verification in electrical engineering to the global community of scientists working in biotechnology. Answering requests and questions from researchers from all over the world about the GOA TOOLs project is an enjoyable and productive way to converse with scientists and to gauge the needs of the community in planning my future work. I am thankful for the opportunity and have enjoyed my time committed to the project.

Dr. Michael Gusenbauer

Dr. Gusenbauer is a researcher in Austria. His interest is in helping researchers to make better use of the exponentially growing scientific knowledge in governing decisions, offshoring, and international management.

I emailed Dr. Gusenbauer after reading one of his compelling papers, and through that year of exploring and learning, created a secret weapon. It is hiding in public as an open-source project found at https://github.com/dvklopfenstein/pmidcite. It is used alongside PubMed to conduct a literature search – a big, beautiful, satisfying, reproducible, informative literature search.

Starting from the early PhD days, the literature search was my least favorite activity in scientific endeavors. And that is a dreadful thing, a major methodological flaw, a dent in my armor – really more of a gaping hole – an admission that no scientist should even whisper.

Using either Google Scholar or PubMed in my old literature searches left ennui in me. They are both top-rated search systems, so why the unrest?

It was a mystery until I happened upon Drs. Gusenbauer and Haddaway's comprehensive paper describing how they evaluated the retrieval qualities of 28 search systems, including both Google Scholar and PubMed [80]. Twenty-eight is a lot; Gusenbauer and Haddaway's project must have been a Herculean effort.

Through their hard work, I found rationalization behind my dissatisfaction with previous literature searches. Reading their paper substantiated that there had been good reason to feel unfulfilled.

Using Dr. Gusenbauer's advice, I sought a closer look at Google Scholar and PubMed. When unsure of something in search, I wrote to Dr. Gusenbauer with many questions. He generously took his time to write back with answers, clearing my confusion. What resulted was new knowledge for me, new literature search methods, a couple of new papers from both of us [105] [79], a

new open-source project from me, and most importantly, the joy of serendipitous findings.

Other researchers likely have similar misgivings of "how searching is just done around here," because Gusenbauer and Haddaway's 2020 paper has received significant of attention, resulting in more than 100 citations in just over a year.

One thing still lingers: why is Google so interested in "court opinions?" What do they do with information regarding court opinions, who accesses it, and when they access it? References to court opinions are all over their documentation. Finding the "contact link" to directly contact Google is a challenge. But there is one link under the heading, "I have noticed an error in a court opinion you are providing. What I can do to help fix it?" This is a direct quote from the Google Scholar documentation accessed on May 5, 2021 [94].

So to that end, I'm extremely grateful for Dr. Gusenbauer's research and time spent in sharing his knowledge with me over the course of this project.

Dr. Dan Diesburg

Dad's death was hard on Mom and me. After he died, I traveled to visit my Mom frequently. And when back at home, I called Mom almost daily, but it did not stop my worry for her. She seemed so alone.

And then she met the most amazing man while they were both doing taxes pro bono for disadvantaged senior citizens and the visually impaired. Mom and Dan were pleased to learn that they both played bridge, so they teamed up

as bridge partners and then happily enjoyed winning numerous tournaments together.

Dan asked Mom to marry him. The wedding was beautiful. She looked lovely in her Sunday church-style, fancy white hat with a wide brim, which was picked out by Dan. They have amazing adventures together, traveling all over the world. He is kind and helpful and terrific company. Mom's dogs love Dan and give him "hugs" all the time. He loves it. His children, who have children of their own, are also sweet people. I do not worry about her so much now.

Dan is the only person in our family who has a PhD. He has been invaluable in offering tips of the trade when it comes to navigating the PhD experience.

Dr. Fred R. Eisner

For years, Fred has been a source of support, encouragement, professional advice, and now, academic knowledge. He is a kind man with a big, loving family, and it shows in his interactions with all the people in the world around him.

His professional experience was invaluable as I prepared for a job interview in New York City (NYC), a quite unusual occurrence as most electrical engineering jobs are not in NYC.

The interviews at the NYC firm, D. E. Shaw Research, were wondrous. I interviewed for a hardware verification position by speaking with members of both the biochemical group and the hardware group. At D. E. Shaw Research,

the biochemical scientists create breakthroughs in molecular knowledge, and the hardware engineers build supercomputers specifically designed to simulate the new molecular knowledge.

Speaking with the biochemists was the high point of the experience and left me wishing that I worked in biochemistry rather than hardware verification. However, at that time, it seemed my path was set forever to electrical engineering.

Fred helped me prepare for a nearly day-long interview with Dr. David E. Shaw, which was surreal and successful. David Shaw had a large office with shelves extending up the high ceilings that were loaded with books. It was the interview of a lifetime, and Fred worked with me so that I could feel ready and enjoy the experience rather than being racked with nerves.

Most recently, Fred supported me in the race to the finish line of my PhD. His knowledge about the process and his uplifting words in times of great stress were a source of comfort and strength and offered a path to a successful outcome. Anyone would be fortunate to know Fred.

Dr. Aydin Tozeren

Dr. Tozeren inspired me to create the techniques that associate various serious human diseases with specific genes. He was excited about the ASCII art that I devised to visualize the associations and encouraged me to proceed further with my techniques.

Dr. Tozeren urged me to participate in a poster contest in the Sidney Kimmel Cancer Center Consortium Symposium. My entry won first place

among the posters for doctoral candidates, and I am grateful for Dr. Tozeren's expert advise.

Carol Berman

Carol Berman has worked with me on all of my papers. She is an excellent writer whose ability to tell a riveting story has led to work in network journalism and corporate communications in Fortune 50 corporations, which are the top 10% of the Fortune 500 companies.

Her challenging open questions felt like they came from a peer reviewer. Her keen observations prompted great reflection and encouraged me to undertake comprehensive investigations. The results were clear and useful conclusions, which were described in my peer-reviewed papers. My papers would not have had that secret sauce without Carol's insightful queries.

Lorraine Peters

Fitness has been an on-going theme in my life, and I am grateful to my Dad for his inspiration. I met Lorraine and Michele, mentioned below, in a rollerblading group on the Jersey shore.

Lorraine Peters is my amazing friend and partner in crime in technological and fitness matters such as attempting to row in a scull, mountain biking, and skiing in big powder out west.

Lorraine is an excellent rower and has gotten me out a number of times into a scull to row on various rivers. Sculls are super skinny and long "row boats" that are as wide as your hips and are super tippy.

Every time I wiped out on my mountain bike on the trail, she would gasp, looking shocked and extremely concerned. “I’m okay. I’m okay. I know how to fall. Relax, tuck and roll. No problem. Let’s keep going.”

We dream and brainstorm together of future projects while attending buck-a-shuck oyster night. Hopefully, we can work together someday on biomedical projects that will change the world.

Michele Maybaum-Harris

The nearly impromptu evenings we had blowing off steam usually turned into adventures. “Hey Michele, want to go to a glo-light fitness party that Nafis is putting on?” Having no idea what that might entail, of course she says yes. We put on glo paint at the party and sweat like a waterfall, taking a break while we watch Jessica get her jawn on slinging the giant viking-style ropes, which are two heavy 2-inch-thick ropes that you hold, one in each hand. The other end of each rope is bolted to the floor twenty feet away. You sling each rope up and down as you are squat real low. It is quite the experience. You should try it. You likely will not be able to do it for too long. Practice, practice, practice.

Jessica Kroboth

Jessica is a card-carrying member of the Nafis glo-light fitness team. In another fitness competition created by Nafis, I competed on a team of three against her and a partner. Her team beat us handily due to her expert fitness prowess.

We did tough Pilates workouts together for most of my graduate years. She was always a kind, supportive listener for my blabbering on about PhD

stresses. I would not have been as sane without her tough workouts to burn the tension and her strong emotional support.

Nafis Austin

Nafis Austin's workouts also kept me grounded. Not being a talkative person in normal life, it is kind of weird how I turn into quite the chatter-bug during a workout.

"Are you injured? Are your knees bad? Why can't you jump up off the floor?" "No I am not injured. This my level of strength. That is all there is." Shaking his head like he'd never seen such a thing, "Oh wow. Well, we'll fix that." And so we did. He had identified a weakness that was previously unknown to me. It is good to identify your weaknesses so you can work on them.

Rosanne Park

Without my childhood friends, Rosanne Park and Sabine Falk, the PhD experience would have been much harder. Rosanne and I brainstormed together about the frustrations of being the only woman in our groups in tech.

During my time in electrical engineering, the highest rank that any woman colleague ever rose to was first-level manager. There were only two of them, each in a different company, and even that did not last. They both were slung back to individual contributors during a re-org.

My super smart friend, Rosanne, soared to much greater heights. But even the power that comes with each successively better position could not entirely

protect her from the brutal tech environment. I am achingly tired of hearing that played-out old trope (and do not you dare ever say it): "Girls just don't like tech." Yes, we do. We like it so much that we fight hard to do the work we love. Rosanne has inspired new dreams in me to contribute to a better future through biotechnology.

Sabine Falk

When we were kids, Sabine and I had long talks on the school bus riding to and from school. It feels like not much has changed. As an adult, Sabine ended up working for IBM for a long time. I hope that we might be able to work together in the future. Talking to Sabine during the PhD times helped keep me grounded.

Christine Macneil

Christine Macneil, who lived on the floor where I was a learning assistant at Rensselaer Polytechnic Institute (RPI) in Troy, New York, U.S.A, has been a friend since that time. We have shared our work experiences as engineers from RPI with one another from the start. She has helped me with the wordings of some of my writings, and she too, has been supportive on calls made after a hard day of research.

Bradley Albright

Most importantly, thank you to Brad, who has been with me through everything: electrical engineering, seriously considering "the switch";

part-time studies in biomedical engineering while working full time; full-time studies in biomedical engineering after quitting my final electrical engineering job; and various pets, such as a street cat that acquired me (and Brad).

After initiating the "big move" from electrical engineering to biomedical engineering by going back to university, he enthused "You go, honey!" and bought me a ton of pens and notebooks . He did not know at that time what he was getting himself into. Me neither.

He championed me even when I was not at my best. I will be forever grateful for his love, encouragement, and support. I could not have done this big switch without Brad.

Table of Contents

List of Tables

List of Figures

Abstract

Protein-coding Hotspots in the Human Genome: Annotation, Significance, and their Conservation in Animal Models (mouse, fruit fly)
D. V. Klopfenstein
Will Dampier, PhD

Investigating understudied genes that are not yet associated with disease but have common functions with nearby genes that are not in the same gene family can lead towards further understanding these molecular mechanisms and may reveal novel drug targets. Previous studies of utilizing population genetics approaches did not focus on the chromosomal topology of many major diseases on the human genome. Clustering algorithms, augmented for this thesis to run on the linear topology of chromosomes in the human genome, identified the densest clusters with ten or more genes, called hotspots. Performing enrichment analysis of the hotspots finds genes which share a genomic hotspot and significant gene functions to highlight genes that are understudied and/or not yet associated with disease. Methods developed for this thesis include new approaches to comparing functions among a set of genes, even if the genes are in different species; a library for examining gene ontology relationships; and an augmented exploratory literature search using PubMed combined with public access citation data from the National Institute of Health's Open Citation Collection (NIH-OCC).

Chapter 1: Illuminating the Druggable Genome

1.1 Introduction

About 3,000 of the 20,000 protein-coding human genes are considered to be druggable [185] [182]. A protein is considered druggable if it has folds that are likely to have interactions with drug-like chemical compounds [182].

For a protein to be considered a favorable drug target, it must first be druggable; second, modifying its biological function must provide therapeutic benefit [182]. Crucial goals in drug development include target identification and target validation. Target identification involves ascertaining the appropriate drug targets for a disease, while target validation involves showing how perturbing the target affects the disease biomarkers and disease phenotype [58]. Of the ~3,000 genes considered to be druggable, only about one in four are targeted by FDA-approved drugs [180].

In addition, only about 10% of all the ~20,000 protein-coding genes in the human genome have been thoroughly studied [180]. A large portion of under studied genes are known to be intriguing to investigate, are associated with human disease, and have important biological processes [182]. Nonetheless, the large majority of human genes continue to be understudied.

Two lists of genes help inform how well a gene is studied. The Illuminating the Druggable Genome (IDG) represents the main list since it rates the coverage of the status of knowledge for all human protein-coding genes. The IDG group separates all human protein-coding genes into one of four categories, ranging from well studied genes targeted by an FDA drug with a known mechanism to genes that are mostly unstudied.

One additional smaller gene list will be used to compare with the IDG list. Stoeger et al. used machine learning methods to study why some genes are highly studied while others are almost ignored and used this information to create a list of 514 understudied genes with GWAS data, experimental data from model organisms, and Genome-wide CRISPR loss-of-function screens. Another interesting list for future tasks is from Wood et al., who researched yeast and found that many conserved genes in yeast that have human orthologs are not well studied. Wood et al. found that genes that are not well studied in yeast tended to be not well studied in humans, supporting Stoeger's same observation.

This chapter describes the breadth of under studied protein-coding genes in the genome and theories of why neighboring genes are co-expressed. It then describes that genes that are clustered together but not found in the same gene family are sometimes functionally related.

1.2 Understudied genes

1.2.1 The Stoeger gene list of 514 understudied genes.

Stoeger et al. created a list of 514 genes considered to be understudied and "experimentally accessible," such as a gene whose loss-of-function perturbs a phenotype and where there is a genome-wide CRISPR loss-of-function screens for the gene. Stoeger et al. consider a gene well studied if it is featured in many publications per year. They used a machine learning model to investigate whether it could predict both the year of first publication for a gene and its total number of publications. Stoeger et al. initially trained the model using

430 physical, chemical, and biological gene features and they found that only 15 features determined most of the model's accuracy. This included items such as the positive charge and hydrophobicity of proteins; the length of the transcript and gene; the presence of signal sequences that encourage the transport of new proteins into the endoplasmic reticulum; and the protein quantity expressed in HeLa cells. Many labs are considering replacing the HeLa cells used in experiments with other cell lines, which may facilitate new discoveries regarding less studied genes since the highly expressed proteins in HeLa may be different in a new cell line. Highly expressed proteins are a crucial feature that correlates to a gene being well studied.

In addition, Stoeger et al. were able to greatly improve their prediction of the number of publications for any human gene by including the year of the first publication for orthologous genes, which they found to be more important for predicting the quantity of publications than the year of the first publication for the human gene. Stoeger et al. thus concluded that knowledge from model organisms propels research on human genes. They noted that the percentage of researchers studying the orthologous genes of model organisms has declined from about 80% in 1970 to about 30% in 2015. There was a brief increase to 40% just before the human genome was completed in 2003, but then it declined to about 30% after the completion of the human genome, while the percentage of researchers studying only human genes increased from about 25% in 2000 to 50% in 2015.

1.2.2 The Wood gene list of ~3,300 understudied genes

Because the date of the first publication describing a gene from a model organism that has a human ortholog is informative to the Stoeger machine learning model, this section discusses Wood's list of understudied human genes with orthologs in fission yeast (Schizosaccharomyces pombe) and budding yeast (Schizosaccharomyces cerevisiae). The ancestors of humans and yeast are separated by a billion years of evolution. Budding yeast and fission yeast are distantly related to one another, with their common ancestor occurring about 330–420 million years ago [192].

The list of ~3,300 proteins was created by the yeast researchers Wood et al., who identified the proteins conserved over one billion years ago when yeast diverged from humans, which intersected with the list of proteins from orthologous genes of fission yeast and budding yeast, which diverged 500 million years ago. Wood's list of common orthologs between fission yeast, budding yeast, and humans contains genes with biological functions that are crucial to all three species, yet they remain in the "conserved unknown" category in yeast research [216]. Because research into model organisms drives research on humans, Wood's list contains genes which warrant further attention.

An example of "conserved unknown" genes in yeast that are orthologous to human genes includes under-studied mitochondrial, internal cell membrane proteins, and nuclear proteins. The under-studied nuclear proteins are found to often participate in scaffolds and molecular complexes [59].

Wood et al. find that about 80% of fission and budding yeast genes have been well characterized, which has remained nearly the same for a decade. Wood states that proteins remain under studied due to bias in biology and research.

Biological biases include favoring studying genes whose loss of function has a clear phenotype in the lab, where yeast grows on nutritious media, and thus any loss of function relating to yeast's inability to grow on rich media becomes readily apparent. Phenotypes observed in the wild but not in the lab, however, can remain unstudied for years, which includes responding to changes in the environment that include processes important in aging, such as protein and fat homeostasis [111], detoxification, and mitochondrial processes. The buildup of damaged or misfolded proteins is exacerbated during aging and can be found in neurodegenerative and motor neuron disorders. Wood et al. emphasize that the ecology of yeast in its natural environment is not well understood, which leads to missing information regarding yeast processes discovered in the lab, such as interacting with infectious agents and insects.

Research biases include researchers limiting their study to a narrow range due to the abundance of targets and a researcher's knowledge in a focused area. Other issues which prevent the study of genes include the lack of research tools such as antibodies, lack of funding, and risks delaying the early career researcher's rise to principal investigator [198] [55]. A smaller percentage of researchers working on model organisms means less research on the orthologous human genes [198] [216]. Basic research on cellular-level processes in model organisms provides a fundamental base for new drug discoveries, therapies, diagnostic testing, and medical procedures.

1.2.3 The Illuminating the Druggable Genome Group

Over 17 years after completing the human genome, nearly one in three protein-coding genes remain understudied according to the group Illuminating the Druggable Genome (IDG), which identifies understudied potential druggable targets for protein superfamilies such as G-protein-coupled receptors (GPCRs), nuclear receptors (NRs), ion channels (IC), and kinases.

The IDG was created to coordinate a large research consortium to study each gene in the human genome by designating each into one of four categories to describe the current state of knowledge regarding each gene and to spotlight genes that are promising drug targets but about which little is known.

Four IDG Categories: Tclin, Tchem, Tbio, and Tdark

The four categories used by the IDG to label human protein-coding genes are Tclin, Tchem, Tbio, and Tdark (Figure 1.1).

Genes labeled with Tclin are the most well studied targets that interact with approved drugs, while Tchem are also well studied targets that interact with approved drugs, but the mode of action is not known. Tchem is also used to label targets that can bind to small molecules with potency that exceeds thresholds set by the IDG.

Tbio targets are associated with a disease in Online Mendelian Inheritance in Man (OMIM) [6], have gene ontology (GO) [10] leaf terms backed with experimental evidence, or meet two of the three remaining criteria: the gene

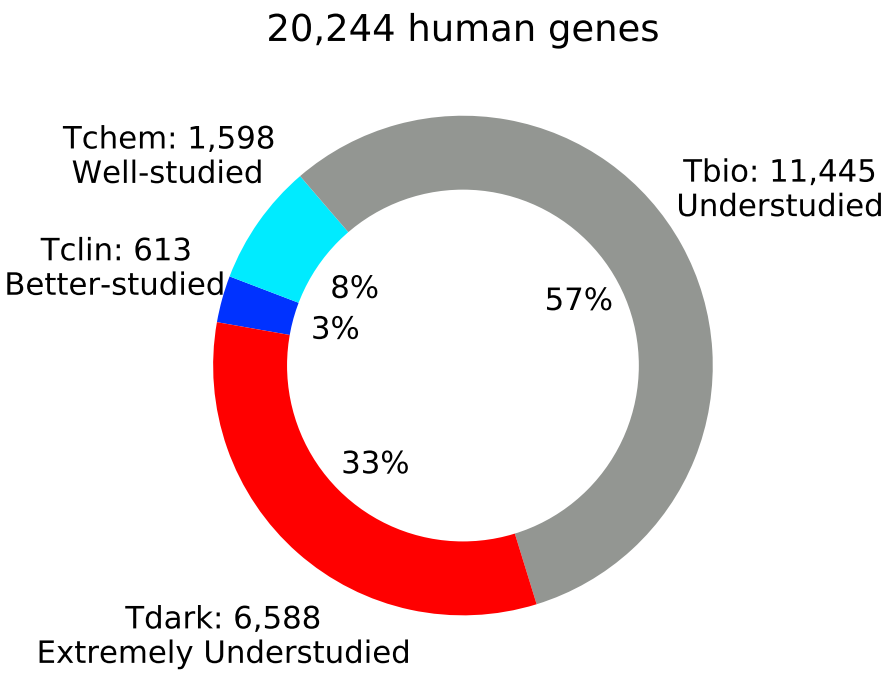

Figure 1.1: **Percentages of genes in the four IDG Categories: Tclin, Tchem, Tbio, and Tdark.** The least studied genes (Tbio and Tdark) comprise 89% of the human genome while only about 11% of genes are well characterized.

has a number of publications which exceeds a threshold; has three or more Gene Reference into Function (GeneRIF) entries; or has more than 50 commercial antibodies, which is tracked in the Antibodypedia database [19].

A GeneRIF is a short phrase describing the function of a gene combined with a reference to a research paper, indexed in PubMed, that describes an experiment which elucidates gene activity [140]. Most GeneRIFs are created by scientists with advanced life science degrees and are employed at the index section of the National Library of Medicine but any researcher can submit a GeneRIF, which is reviewed before being published.

The last category, Tdark, marks the genes about which the least is known. The primary sequence of Tdark genes must be manually curated and available in UniProt. A gene is Tdark if other information about it does not reach the level of a Tbio gene. Dark genes also lack many analysis tools such as antibodies, which can help identify where a protein is expressed in the body [55].

The set of Tdark genes, which is about one in three human protein-coding genes, is called "the dark genome." This includes genes and proteins where little is known regarding biological function, which hinders understanding the effect of human genetic variation on disease [25].

Only ˜11% of genes are well characterized (Figure 1.1), are labeled Tclin and Tchem, and are 3% and 7.9% of the genome respectively. Enzymes, transcription factors, and kinases are the most abundant gene families in the human genome (Figure 1.2). Transporters, olfactory GPCRs, non-olfactory GPCRs, and ion channels are the next most abundant and account for 473, 421, 408, and 344 genes respectively. Sixty percent of gene families are denoted as non-IDG, indicating that the IDG has not specified their gene family.

Examining the distribution of gene families across TDL groups shows that both enzymes (Figure 1.3 yellow bars) and non-IDG genes (Figure 1.3 blue bars) appear in each TDL category. Due to the great difference in percentage between Tclin (3%) and Tdark (32.5%), it may appear that there are only a small number of Tdark enzymes; however, because only 11% of genes are Tclin or Tchem and 32.5% of genes are Tdark, the numbers of enzymes in Tclin and Tchem are about the same as those found in Tdark, which is around 750 (Figure 1.4). There are ˜2,700 enzymes in the Tbio category. The Non-IDG gene

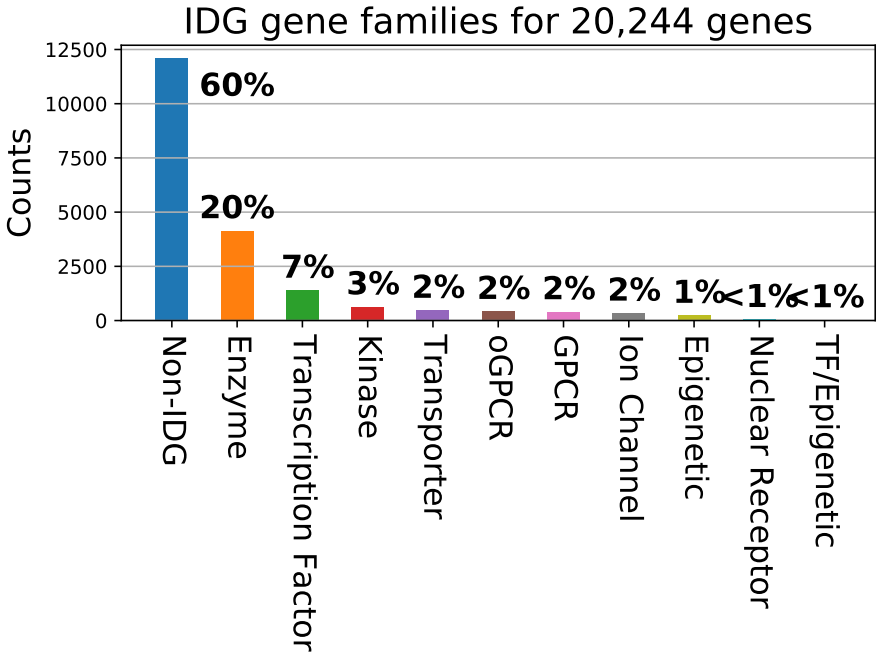

Figure 1.2: **Percentages of gene families in the human genome.** The largest gene family in the human genome is enzymes.

families also appear across all four TDL categories, and although their counts are small in the Tchem and Tclin, they are in the thousands for both Tbio and Tdark.

Examining the percentages of TDLs in each gene family shows trends regarding which families are well-studied.

1.2.4 Why do genes go unstudied?

Before the completion of the human genome in 2003, only about 10% of proteins were well studied. In 2011, the same genes mostly remained well studied, while much of the rest of protein-coding genes were ignored, even when linked to disease [55]. Stoeger et al. found that 49% of the publications in

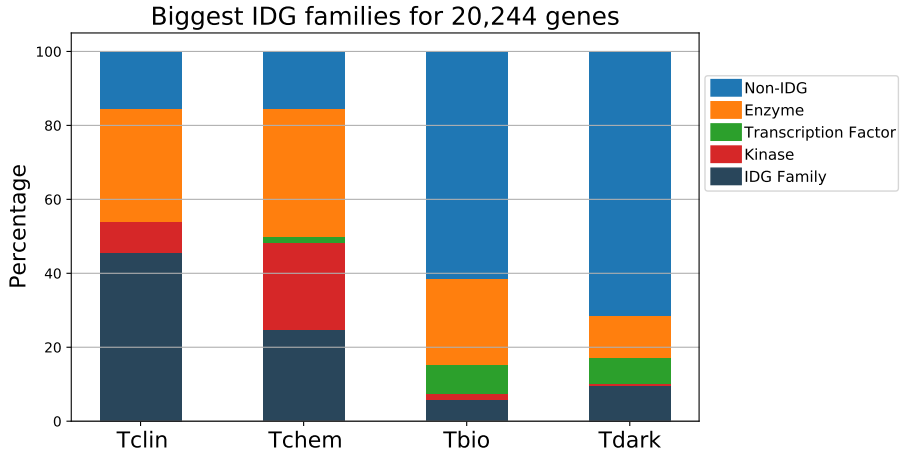

Figure 1.3: **Percentage of gene families in each TDL category.** The largest gene family in the human genome is enzymes, whose genes are seen across all four TDL groups. Kinases are also seen across all four TDLs. Enzymes and kinases account for almost 40% of Tclin genes and 58% of Tchem genes.

2015 discussed the small group (16%) of genes described in publications from 1991 [198].

One supposition for concentrating on a small portion of genes is that research focuses on the important genes, thereby intentionally leaving the unimportant genes understudied [198], which has not proven to be true.

Using a small number of gene properties, Stoeger et al. found that they could predict the approximate year of publication for the first research paper describing a specific gene, the number of papers for a human gene, the amount of National Institutes of Health (NIH) funding awarded to study the gene, and the level of drug development against disease-related genes.

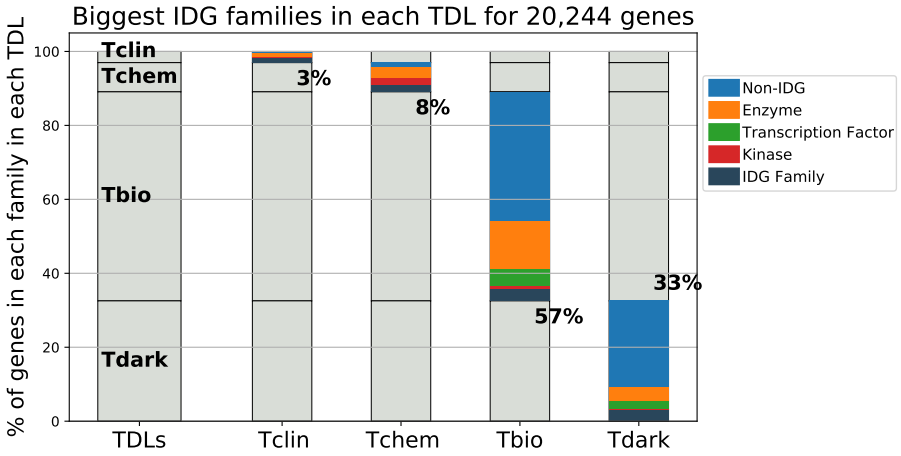

Figure 1.4: **Percentage of gene families in each TDL category.** The largest gene family in the human genome is enzymes, whose genes are seen across all four TDL groups. Kinases are also seen across all four TDLs. Enzymes and kinases account for almost 40% of Tclin genes and 58% of Tchem genes.

Nature of understudied Genes

Researchers mostly focus on the following gene categories: virtually all carbonic anhydrases and phosphodiesterases, which are enzymes, are well studied, and have FDA-approved drugs which act upon them; nuclear receptors are well studied categories of molecules and have no Tdark genes; although non-olfactory G-protein coupled receptors (GPCRs) are well studied, about half are still categorized as under studied (Tbio and Tdark).

The categories of understudied genes are discussed below. Oprea found that nearly all the ˜420 olfactory GPCRs are Tdark. Other categories with no or few FDA drugs which act upon them include transcription factors, solute carrier transporters, and enzymes including transferases, phosphatases, and small GTPases [156].

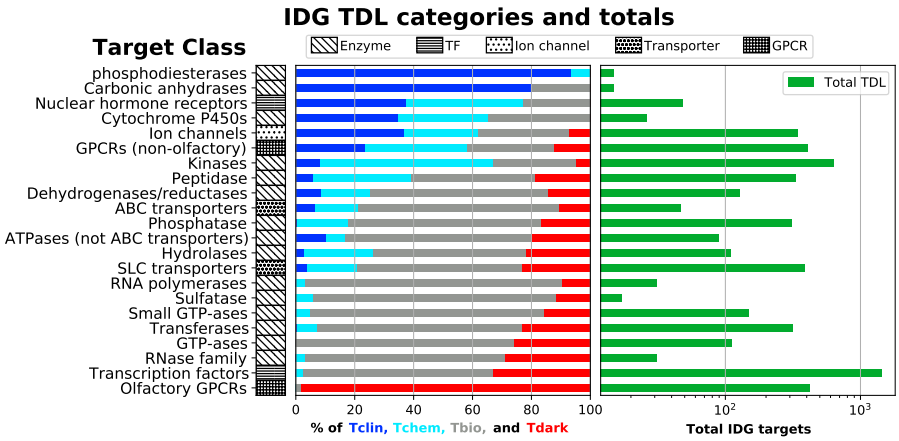

Figure 1.5: **Percentage of TDLs for each gene family.** Phosphodiesterases are the most well studied gene family but contain only 15 genes. Transcription factors have no Tclin genes and only 36 Tchem genes. The remaining 97% of the 1,438 transcription factors are Tbio and Tdark.

Stoeger et al. studied why only about 2,000 genes out of ~20,000 were well studied by examining hundreds of gene features. Using about fifteen biological, chemical, and physical properties of a gene plus the level of research devoted to homologous genes on model organisms, they could predict the year of the first publication, the number of publications, the level of National Institutes of Health (NIH) funding, and the development of drugs that act on the gene products.

Funding represents a reason that genes which are promising drug targets remain under studied, because grants are frequently granted to research well studied genes to study a new hypothesis[157]. Pharmaceutical companies mitigate risk by developing drugs targeting well studied genes [182].

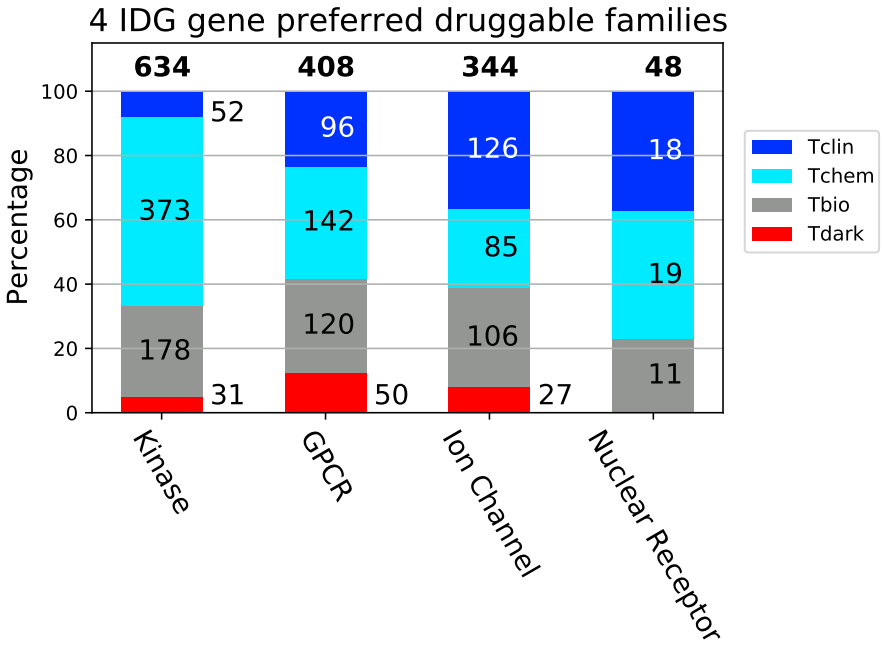

Figure 1.6: **Almost 40% of the four IDG-preferred druggable families are understudied.** The upper dark blue and cyan bars represent well studied genes. The lower grey and red bars represent poorly studied genes.

Model Organisms

Knowledge from studying model organisms drives research on human genes. Stoeger et al. found that 50% first publications report on human genes. The percentage of scientists studying only non-human genes fell from over 80% in the 1970s to about 35% in the mid 2010s. The percentage of scientists studying only human genes rose from 20% in the 1970s to about 45%, with a jump in 2000 just before the human genome was completed and another rise about 2005 after the completion of the human genome [198].

Even among well studied eukaryotic model organisms, only about 80% of proteins have well characterized biological processes [216]. This number has only increased from 77% since 2007 [161] for budding yeast.

Wood et al. used GO annotation to create criteria to show the level of functional annotation on a proteome, which regards a widespread method of describing the gene product function for a multitude of species using three branches: biological process (BP), molecular function (MF), and cellular component (CC).

The GO BP terms describe overall processes ("biological programs"), which are comprised of numerous molecular activities. The GO BP is not the same as a biological pathway, although pathway databases such as Reactome annotate GO BP terms to their pathways, MF terms to their reaction events, and CC terms on both pathways and events.

Pathway databases contain the dynamics and dependencies required to fully describe a biological pathway.

The GO MF describes the chemical reactions and activities performed by individual gene products or molecular complexes containing multiple gene products.

The GO CC describes the locations related to the cellular structures where a gene product performs a function of the GO BPs used in pathways and gene products used in chemical reactions. Complementary experimental techniques are necessary for the GO branches. Wood et al. consider a gene to be well characterized if a gene product was annotated using GO terms from all three branches, and they consider a gene product to be uncharacterized if no GO terms are annotated to a gene product.

1.2.5 International Mouse Phenotype Consortium

The IDG works with many large programs to study the dark genome. Partners include the Structural Genomics Consortium (SGC), whose mandate is to determine the 3D structures of human proteins with biomedical importance. The SGC deposits between 25% to 50% of all structures in the Protein Data Bank (PDB) every year.

Another large collaborating partner of IDG is the International Mouse Phenotype Consortium (IMPC) [143], which investigates IDG's list of Tdark genes by creating knockout mouse strains for each gene.

One method of exploring Tdark genes is to compare human diseases with the phenotypes of experimental animals such as mice. Using both mouse and human phenotype information strengthens the evidence from each species.

The list of genes prioritized as Tdark by the IDG group are sent to experimental centers such as the IMPC [108] to prioritize developing embryonic stem (ES) cells and knockout mice that have a significant change in phenotype. It is this prudent to check for mouse phenotype data for a gene set.

1.3 Gene neighborhoods

1.3.1 Regulatory elements

Regulatory elements for gene expression include transcription factors, enhancers, promoters, silencers, chromatin remodeling, repressors, and the three-dimensional chromatin topology. The speed at which the regulatory elements can respond to changes in developmental or environmental stimulus also affects gene expression.

Discoveries are changing beliefs about gene regulation. For example, the functions of promoters and enhancers were believed to be distinct and separate until recently. Promoters define the location of transcription initiation [113], while enhancers amplify transcription initiation [191]. It has been recently found, however, that some promoters have enhancer behavior [46] [41] and that some enhancers have some promoter activity [8]. New technologies drive the new discoveries of promoter and enhancer functions. Massively parallel reporter assays were mentioned in the literature in 2012, but mentions have more than doubled since 2017. CRISPR–Cas9 developments have helped researchers understand the function of targeted regulatory elements. Cas9 in CRISPR can be used for gene activation and repression using a nuclease deficient Cas9, which does not cut the DNA but instead can bring repressors or activators close to the area of interest when fused with Cas9.

New discoveries regard pathways as well as regulators. Until recently, plant metabolic pathways were considered to be mostly composed of genes scattered throughout the genome. Metabolic pathways create metabolites, which are estimated to number over a million in plants [154] and combat pests and competing plants, attract pollinators, and configure the microbiome. Plant products are used in medicine, industry, and agriculture [154]. In 1977, the first metabolic pathway composed of genes clustered together was discovered [65]. By 2012, a total of nine metabolic pathways would be known to have clustered genes. In 2014, it was recognized that most metabolic pathways in plants are not yet discovered, nor has their configuration in the genome been determined [154]. In 2018, it was recognized that clustered genes in plants allowed the plant to react to environmental changes. The honeybee's significant plasticity

in terms of oogenesis has been determined to originate from gene products which appear in the same pathway in which all are clustered.

In 2020, Liang et al. found that in keratin clusters expressed during the development of a chicken embryo, an epidermal cell developed into either a scale on the foot or a feather on the body due to the local co-regulation of genes in a gene cluster. An epidermal cell fated to become a feather, however, became either a downy feather or a flight feather due to genes located far apart but then brought together for co-regulation using chromatin loops [120].

1.3.2 Gene cluster history

Gene clusters are comprised of genomically co-localized and possibly coregulated genes that are often found to be conserved across species [130].

Stoeger and Woods observed that researching genes in model organisms that have human orthologs drives research on human genes. It is important to review the history and recent discoveries concerning how the linear genome architecture in flies, bees, yeast, and plants impacts biological processes. This can often be overlooked by researchers of human biology since many researchers tend to research in their own area. Plant papers are rarely cited by non-plant papers but may provide important insights for non-plant researchers. Figure 1.7 shows a timeline of gene function similarity in clusters discovered in various species.

It was found that housekeeping promoters cluster together in flies (Drosophila), enabling genes to be co-expressed. The co-expressed genes were found to be housekeeping genes, which are expressed in all tissues [37]. In 2003, it was believed that only housekeeping genes are clustered together

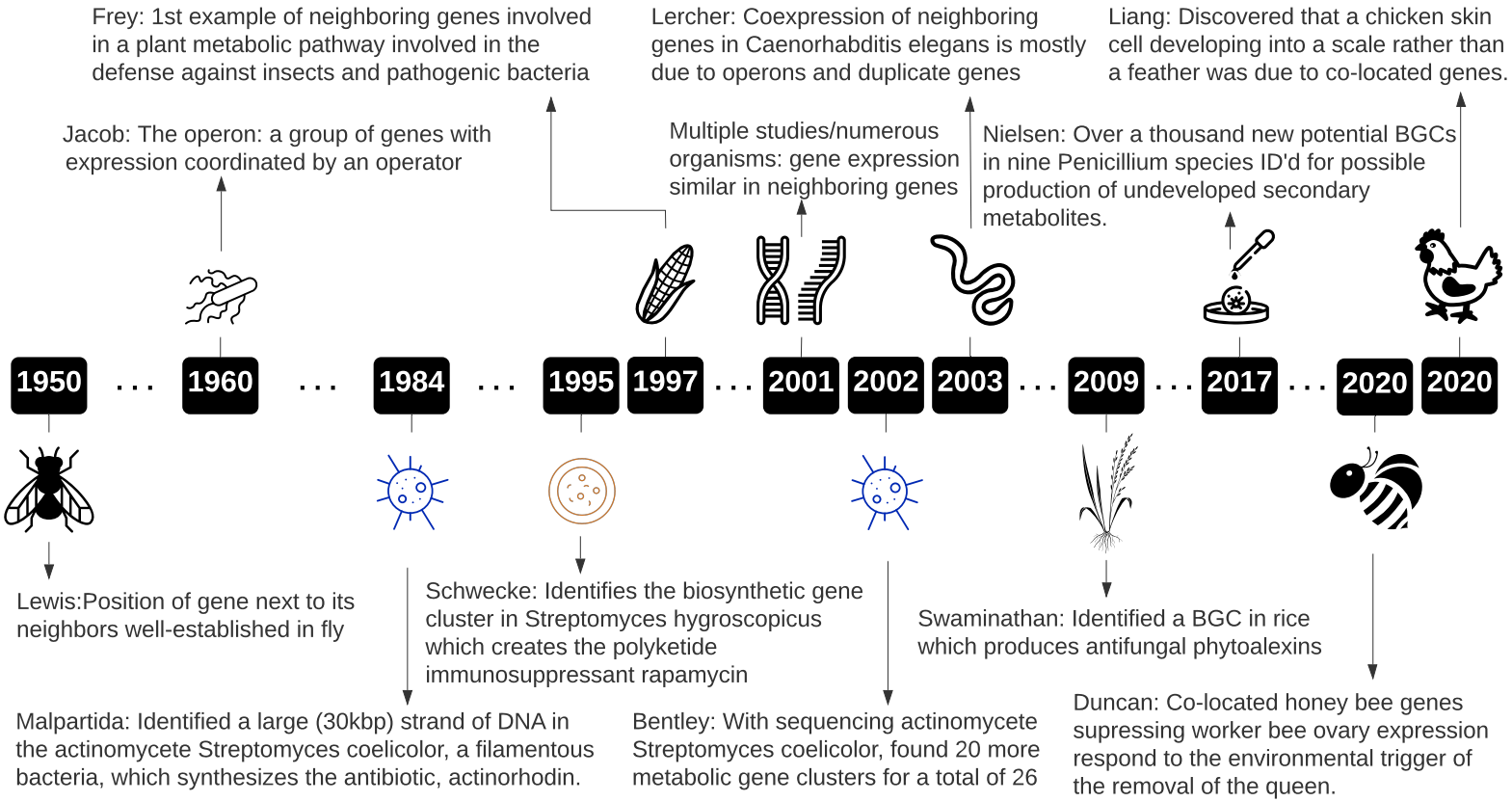

Figure 1.7: **Timeline of gene function discoveries in gene clusters on chromosomes.** Over time, researchers have learned more about how genes in a cluster can have similar functions or be involved in similar processes.

while tissue-specific genes are scattered around the genome [110] but new evidence shows numerous cases in which the genes in the same pathway are clustered together [153]. Other researchers have found that co-expressed genes are usually not functionally related [71].

Plant researchers have discovered new natural product pathways where the pathway genes were clustered together in the genome [153]. These new discoveries have been enabled by the falling cost of sequencing. Honeybee researchers have discovered the drastic phenotypic change of remodeling the ovarian tissues of the worker bees, which do not produce eggs, into a potential queen bee that creates and lays mature eggs, which results from gene products whose genes that cluster and are regulated together in the genome [52].

Lewis, 1950: The position effect in fly

In 1950, Lewis noted that it is established that the position of a gene in relation to its neighbors can affect the phenotype of a fruit fly [115]. Lewis cited Sturtevant's 1925 investigation concerning the variations of the shape of fruit fly eyes, which ranged from large and round in normal fruit flies to narrow and slit- or vertical-bar-shaped in mutants. These effects were observed due to the bar gene being duplicated or tripled in the mutant fly, which is caused by crossovers occurring near the bar gene during meiosis. Using over 100,000 files in his experiments [215], Sturtevant observed that the mutant eye-slit was enhanced (made narrower) if there were three bar genes on one homologous chromosome (mutant) and one bar gene (normal) on the other homologous chromosome (3/1 bar genes) compared to two bar genes on each homolog in the homozygous chromosomes (2/2 bar genes). Both examples (3/1 bar genes

and 2/2 bar genes) have four copies of the bar gene, but the fly with 3/1 bar genes had a more pronounced mutant slit or bar-shaped eye compared to the fly with 2/2 bar genes. This evidenced that it was the cis or trans position of the bar genes that affected the phenotype, the fly with a triplicate of bar genes had a greater effect compared to two duplicates of bar genes. He thus concluded that it was the position of a gene next to its neighbors that produced a difference regarding phenotype or function [199].

Since Sturtevant's 1925 paper, in 1950 Lewis noted additional researchers who found that the position of a gene next to its neighbors had effects on fruit flies, such as increasing the hairs on the back of the fly.

Jacob, 1960: The operon in bacteria

In 1960 while studying the bacteria Escherichia coli, Jacob and Monod discovered the operon: a cluster of co-linear genes whose expression is coordinated by an upstream operator in response to changes in the environment [96], for which they won the Nobel prize in 1965 [116]. Glucose is the preferred food of E. coli, and if glucose is not available, E coli will consume another sugar (lactose). This rapid adjustment to food occurs in its environment using the operon. The genes in the lac operon are lacZ, lacY, and lacA, which lie close together co-linearly and just downstream from the operator and the promoter, and they are transcribed together to create a single mRNA. The first step in lactose metabolism, cleaving the lactose into galactose and glucose, is accomplished by the protein product of the lacZ gene called beta-galactosidase. The protein product of the next gene required in consuming lactose, lacY, is lac permease, a transmembrane protein which aids

the lactase when entering the cell. The protein product of the next gene, lacA, is transacetylase and helps the viability of the cell by moving an acetyl group from coenzyme A (CoA), a molecule important in generating energy, to the hydroxyl group of glycosides containing galactose. In the presence of E. coli's preferred food, glucose, a repressor molecule tight to the operator just upstream from the lacZ gene, thereby repressing the proteins discussed above. The absence of glucose and presence of lactose will cause the repressor molecule to release from the operator and the lacZ, lacY, and lacA genes will all be expressed together, causing E. coli to consume lactose. The important innovation from Jacob and Monod was that metabolism and gene expression could rapidly respond to the environment and that genetic organization, specifically the set of co-linear lac genes, was crucial to lactose consumption.

Malpartida, 1984: Identified a Biosynthetic Gene Cluster (BCG) in bacteria

In 1984, Malpartida and Hopwood isolated a large (30 kilobase pairs (kb)) continuous segment of DNA in the Streptomyces coelicolor (a bacteria that lives in the soil, decomposes dead organisms, gives the soil its earthy smell, and does not cause disease in humans, plants, or animals) that contains all the genes needed to synthesize the antibiotic called actinorhodin. Inserting the isolated Streptomyces coelicolor DNA into Streptomyces parvulus gave the host bacteria all the genes necessary to create the antibiotic actinorhodin [129]. Most antibiotics produced today come from the Streptomyces genus [88].

What Malpartida and Hopwood called "a large continuous segment of Streptomyces coelicolor DNA which apparently carries the complete genetic information required for synthesis of an antibiotic" [129] in 1984 has become

known a biosynthetic gene cluster (BGC) starting in 1988. BCGs evolve quickly and often comprise a complete pathway that produces a small molecule that is amenable to identification through assays [63]. The number of papers with "biosynthetic gene clusters" in the title or abstract have dramatically increased since a few papers published per year until 1997 to over 250 papers published per year in 2020.

Gene clusters for secondary metabolic pathways have long been known to be in bacteria and filamentous fungi. Examples were discovered in plants just before 2010 [158].

Schwecke, 1995: Identifies a BCG that produces immunosuppressive molecules in bacteria

Rapamycin was discovered in 1975 [209] when analyzing Streptomyces hygroscopicus bacterium found in a soil sample from Easter Island by the gastroenterologist Stanley C. Skoryna in 1964, who noticed that the residents did not become infected with tetanus caused by the soil-dwelling Colstridium tetani bacteria entering the body through a cut in bare feet [50]. Tetanus is a bacterial infection frequently found in locations with horses, and Easter Island had more horses than people in 1964.

In 1975, Vézina et al. were able to extract, isolate, and characterize the structure of the antifungal molecule, which was labeled rapamycin, after Rapa Nui, the name that the indigenous people gave to Easter Island [209]. In 1977, Martel and Galet discovered that rapamycin inhibits the immune response and is an antibiotic [131]. In 1995, Schwecke et al. identified the biosynthetic gene cluster that produces rapamycin [188].

The Streptomyces hygroscopicus bacterium produced an antifungal antibiotic (rapamycin) that was effective against Candida albicans, which is a fungus that lives in mucosal tissues such as the mouth and intestine. It can cause a systemic fungal infection with a mortality rate between 25% and 30% in cancer patients receiving chemotherapy due to the commensal bacteria losing their ability to regulate the resident Candida albicans [18]. Candida albicans also cause thrush, which manifests as white lesions in the mouth.

Rapamycin is used to prevent the rejection of implanted organs and, in 2008, was proposed as an anti-aging drug using doses lower than those used to prevent the rejection of implanted organs [20]; however, rapamycin can have severe side effects that must be monitored, such as early-onset diabetes, cardiomyopathy, renal, and liver issues [148].

Frey, 1997: BGCs are found in grasses and corn

By 1997, biosynthetic gene clusters were shown to frequently appear in bacteria, when an initial example of neighboring genes involved in a metabolic pathway was discovered in plants: a pathway involved in the defense against insects and pathogenic bacteria in grasses and corn.

Lercher, 2002: New sequencing technology expands pathway knowledge in bacteria

New technology aided finding new biosynthetic gene clusters. When the complete genome of the soil-dwelling bacteria Streptomyces coelicolor was sequenced in 2002, 20 more metabolic gene clusters were found for a total of 26 known BGCs [17].

Lercher, 2003: Operons in worm are co-expressed

In 1993, Spieth et al. found operons in Caenorhabditis elegans [196], and in 2003, Lercher et al. confirmed that genes in C. elegans operons were co-expressed [114]. Lercher also found that duplicate genes were co-expressed in worm.

Swaminathan, 2009: Found BGCs in rics

Despite the BGC found in grasses and corn in 1997, it was still considered unusual to find BGCs in plants, even in 2009 when Swaminathan et al. identified a BGC in rice which produces antifungal phytoalexins [203]. In 2009, Mugford et al. identified a BGC in oats which synthesizes avenacin, an antimicrobial that provides disease resistance to the plant [142]. By 2015, more plant clusters were found until it became recognized that BGCs are not solely in bacteria and fungi but also in plant genomes. Because information about BGCs was scattered, the Genomic Standards Consortium, formed in 2005 with the aim of making genomic data discoverable, created the Minimum Information about a Biosynthetic Gene cluster (MIBiG) in 2015 and proposed a data standard for describing gene clusters and repository where information about each cluster can be stored (`https://mibig.secondarymetabolites.org/stats`) [138]. Minimum information about a BGC includes key publications, nucleotide accession and coordinates, and the final compound's name, structure, and activity [138]. They mined the literature to find 1,170 gene clusters, which were experimentally characterized, to be the initial BGCs in MIBiG. Researchers for each pathway

were invited to participate by adding missing data about their pathways and more detailed information when possible. In 2020, MIBiG 2.0 has 1,923 clusters with 1,434 minimal entries, 489 non-minimal entries, 465 complete entries, and 27 incomplete entries [77].

Nielsen, 2017: Using new sequencing technology greatly expands BGCs in fungus

Noting that BGCs were mostly studied in bacteria, Nielsen et al. sought to explore the pathways that produce secondary metabolites in Penicillium, a filamentous fungus. They sequenced the genomes of nine species of Penicillium to identify 1,317 putative BGCs.

Duncan, 2020: Gene expression from gene clusters seen in phenotypic plasticity in honeybee

To study the genomic architecture in pathways, which enables an organism to change its phenotype in response to the environment, called phenotypic plasticity, Duncan et al. chose to study the reproductive plasticity of the honeybee (Apis mellifera) [52]. The queen bee is the egg layer while the worker bees tend to the hive and make honey. The worker bees have ovaries but do not lay eggs since their ovaries are induced into a quiescent mode due to the presence of the queen mandibular pheromone (QMP). When the queen dies, the QMP is no longer produced and the worker bees' ovaries begin to produce eggs. Duncan et al. hypothesized that the genes needed for the phenotypic change in the worker bees' ovaries from quiescent to egg-laying must lie together in a gene cluster in order to enable the genes' necessary expression to

be coordinated together. They compared the gene expression in three kinds of bees: queen bees, worker bees with queens in the colony (queen-right), and worker bees without queens in the colony (queen-left) to find that 2,912 genes were differentially expressed, 35% of which were found in gene clusters ranging in size from three to nine genes.

Liang, 2020: Discovered clustered genes important in chicken feathers vs. scales on feet

Liang et al. discovered that a chicken skin cell developing into a scale rather than a feather was due to co-located genes. Liang found that keratin gene clusters in chickens (Gallus gallus) were responsible for the development of chicken feathers versus the scales on the chicken's feet, while 3D chromatin looping is responsible for determining whether a chicken feather would become a downy or flight feather [120].

1.4 Conclusion

Information about gene clusters is rapidly changing. It was once thought that gene clusters were mostly important in bacteria, but then discoveries were made in fungus and soon followed by BGCs found in plants, mostly discovered after 2010. In 2020, there is evidence of gene clusters having a phenotypic effect in honeybee and chicken embryos. Liang et al. made such discoveries in clusters of keratin genes, which are mostly "dark" genes according to the IDG. Liang et al. said that their comprehensive study on chicken epidermis could benefit from new technologies such as CRISPR/Cas9

systems to further inform their work. Genomic architecture, especially gene clusters, are worth further study since there will be much to discover regarding gene function and biological phenotype.

Chapter 2: New literature search method

2.1 Motivation

My first literature search for gene neighborhoods completed at the beginning of the project using PubMed and Google Scholar delivered few results. This sparked the notion that the PubMed experience and results could be easily improved if free citation data were easily accessible. The Google Scholar results could not be improved from outside of Google.

I completed another for research papers about gene functional similarity using Google Scholar, which also yeilded limited results. but remained hard to quantify.

In Fall 2019, three events allowed a targeted comparison of Google Scholar and PubMed and potentially improving the PubMed search experience. The first event was the preprint of Gusenbauer and Haddaway's study, which used 27 search criteria to evaluate 28 search systems, including Google Scholar and PubMed [80]. The second event was the publication of a paper announcing free, easily accessible citations and translation prediction data from the NIH [92]. The third event was the release of "The New PubMed" [127].

The Haddaway guided about how to compare search systems from experts in the bibliometric field. Haddaway's 27 search criteria were based on five foundational search criteria devised by Boeker et al. [22] in 2013 when they studied the recall and precision of 14 systematic reviews using Cochrane systematic review strategies translated to the Google Scholar search tools. Comparing Google Scholar and PubMed provided conclusions using the five Boeker criteria. The frustrations of using Google Scholar in my first successful

gene functional similarity search corresponded with Haddaway's conclusion that Google Scholar was an inappropriate primary choice for certain types of searches. Second, the PubMed search experience could be improved by mimicking the effects of using an immensely popular Google Scholar feature, the *Cited by N* link, by downloading and reporting NIH citation data and relative performance of publications chosen by the researcher. Researchers could combine PubMed results with NIH citation data using a simple command-line tool intended for use alongside the PubMed web interface.

I created the command-line tool and tested its use by writing a peer-reviewed paper about bibliometrics due to the potential improvements to my exploratory literature searches. The results were so successful that the editors of the journal, *Research Synthesis Methods*, made my paper a "discussion/commentary" format, which is reserved only for authors who have been invited to submit. The editors invited Gusenbauer and Haddaway to respond, and thus my publication became part of a three-paper discussion that included the original Gusenbauer paper, my paper, and finally the Guesenbauer response.

2.2 Abstract

I read with considerable interest the study by Gusenbauer and Haddaway (Gusenbauer and Haddaway, 2020, Research Synthesis Methods, doi:10.1002/jrsm.1378) comparing the systematic search qualities of 28 search systems, including Google Scholar (GS) and PubMed. Google Scholar and PubMed are the two most popular free academic search tools in biology and

chemistry, with GS being the number one search tool in the world. Those academics using GS as their principal system for literature searches may be unaware of research which enumerates five critical features for scientific literature tools that greatly influenced Gusenbauer's 2020 study. Using this list as the framework for a targeted comparison between just GS and PubMed, I found stark differences which overwhelmingly favored PubMed. In this comment, I show that by comparing the characteristics of the two search tools, features that are particularly useful in one search tool, but are missing in the other, are strikingly spotlighted. One especially popular feature that ubiquitously appears in GS, but not in PubMed, is the forward citation search found under every citation as a clickable *Cited by N* link. I seek to improve the PubMed search experience using two approaches. First, I request that PubMed add *Cited by N* links, making them as omnipresent as the GS links. Second, I created an open-source command-line tool, *pmidcite*, which is used alongside PubMed to give information to researchers to help with the choice of the next paper to examine, analogous to how GS's *Cited by N* links help to guide users. Find *pmidcite* at `https://github.com/dvklopfenstein/pmidcite`.

2.3 Introduction

Modern literature reviews are primarily performed using online search engines [84] [51]. The two most popular free academic search tools that are commonly used in health studies are PubMed and Google Scholar (GS)[149]. Researchers worldwide are drawn to GS as the most common starting point for literature searches [84] [149] [13] [151] because of its intuitive and familiar search

interface [84] [70], a forward citation search *Cited by N* link under every document result, a *Cite* link to download a document's citation to bibliographic management software such as EndNote for every document result, high citation counts, immense literature coverage, and researcher profile pages. GS's massive citation count, reflected in the "N" in their *Cited by N* links is due to their highly effective web crawlers and to agreements with publishing houses (S3 Figure 13).

But the GS search interface has severe deficiencies that make literature searches laborious and, most importantly, unreproducible. However, many researchers are unaware of the drawbacks of GS [22]. For example, search results for a given query are dropped from one month to another [149] [80] [213] [97], with no documentation as to what has been dropped. Additionally, there is no way to download full search results in bulk [22] [80], resulting in the need to click and click and click to page through up to a maximum of 1,000 search results, 10 or 20 results at a time (S3 Figure 12). And there is no direct access in GS to a paper's digital object identifier (DOI), which is a unique standardized persistent identifier.

It is important to call out the features and shortcomings of both PubMed and GS following two recent events. First, the National Institutes of Health (NIH) published a paper on October 11, 2019 announcing the NIH Open Citation Collection (NIH-OCC), a free public citation database with citation data available for download in bulk [92]. Citation records in the NIH-OCC database are accessible through a set of web and Application Programming Interface (API) tools, collectively called "iCite." Second, on November 18, 2019 the U.S. National Library of Medicine (NLM) announced that the new PubMed

[62] [60] was available [127]. Highlights of the new PubMed include: a nimble mobile experience from a single responsive website for all screen sizes including mobile phones, tablets, and desktop computers; faster and more comprehensive search response; and advanced search features that GS simply lacks.

While I argue that PubMed is superior to GS in many ways, there is room to improve the literature search user experience in PubMed. I compare the "forward citation search" implementation in GS to that of PubMed, finding that the PubMed user experience can be improved by adding a GS feature to the PubMed Graphical User Interface (GUI). Alternatively, command-line users can immediately augment their PubMed search results using the *pmidcite* scripts and library, which download citation data from the NIH-OCC database using NIH's 'iCite" API.

2.4 Scientific search interface requirements

Many researchers are unaware that there is more than one type of search [212] [12] [80] [85], with each search type oriented to different user goals. All search tools are not appropriate for all search types. Three types of search include lookup tasks, exploratory search, and systematic search.

Lookup tasks are the most basic kind of search, usually involving a single query to obtain a well-defined result. An example of a lookup task is searching for a specific paper by entering its title into the query box. GS excels at returning papers when provided with their title, even if there are errors in the query title text. Fellow researchers have complained that PubMed will

sometimes not find a title if it is spelled incorrectly. Gehanno et al. found that 100% of the 738 papers in their study were found using GS to search for each paper by entering its title into the query box [68]. From this, they concluded that GS could be used in systematic reviews [68]. This conclusion was quickly disputed by Giustini and Boulos whose paper is titled, "*Google Scholar Is Not Enough to Be Used Alone for Systematic Reviews*" [72].

Exploratory search and systematic search are useful in evidence synthesis. The goal of exploratory search is the acquisition of new knowledge and is considered to be demanding and potentially time-consuming for the researcher [12]. A researcher doing an exploratory search uses a number of queries to iteratively learn about a subject. The queries begin with a rudimentary understanding of the subject matter and become honed as the researcher's knowledge increases through the search process [212]. Search tools like Google are frequently used in exploratory searches because they are made to be "user friendly" to increase user engagement, which benefits Google by making their market bigger [80]. Google, with their user-friendly interface nearly always returns search results, but the search results that are missing can not be known. Additionally, GS is not designed for systematic searches where researchers need control over the selection power of the query results.

Systematic search is profoundly different than exploratory search. The goal of systematic search is to catalyze an objective account of the cumulative state of evidence for a specific research question. An example research question is "What is the best treatment for lupus nephritis that was classified as stage IV on a renal biopsy [85]?" A well-founded question addresses a clinical need where there is uncertainty regarding the effects of different interventions,

which may vary in practice [85]. The goal is to understand the costs and benefits of various treatments, so that together the doctor and patient can make an appropriate choice for their particular situation. Systematic reviews are an exacting evidence syntheses featuring numerous rigorous steps documented in method guidelines [66] with the goal of providing an exhaustive synthesis of a well studied area of research [80].

Cochrane is one of the organizations that participate in systematic reviews [80]. Steps in a Cochrane systematic review include creating the research question, building a team of people that includes those that have previously done a systematic review, writing or updating a protocol for the review, and having the protocol reviewed. Only after those steps, the systematic search using search tools begins by attempting to find all published and unpublished literature that may answer the research question. First and second authors work independently to remove irrelevant results and upon completion, compare their findings. Many other steps occur, which are all reviewed, under the umbrella of data synthesis and specialized plots, data interpretation, and data presentation. Finally, the review is written. To learn more, I recommend researchers read "How to write a Cochrane systematic review" [85].

High quality literature searches, both systematic and exploratory, are one of the important elements required for the creation of sound scientific evidence [137]. In late October 2013, Boeker et al. recommended that a scientific search interface contain five integrated search criteria. The 2013 Boeker guidance greatly influenced the Gusenbauer study [80], which expanded the Boeker list from five search criteria to twenty-seven for their study of twenty-eight search tools. The requirements for search interfaces are mandatory not only for

structured scientific literature retrieval like systematic reviews, but also in any research that needs to provide a comprehensive literature review [22]. I add "Forward citation search" to the Boeker list to evaluate the extremely popular GS implementation of this feature against the PubMed implementation and compare PubMed and GS's support for the search tools below using the 2013 foundational Boeker advice [22]:

- **Reproducible search**: A reproducible search is a critical quality measure of a systematic review in a well-documented search process that allows others to replicate or update a published synthesis search. To be a reproducible search means that given a search, the same query returns the same results plus new results.
- **Export**: Users should be able to export search results in full.
- **Search history**: Histories are needed to create incremental search changes, which are used to selectively focus the search results.
- **Search strategy documentation**: Documentation that instructs researchers how to create original search queries and how to iteratively develop new queries that build upon previous searches.
- **Search string builder**. These include the use of numerous fields, such as author, title, journal, date, and abstract, and clinical query filters for categories including therapy and diagnosis.
- **Forward citation search**. Tools that allow researchers to follow the chain of citing papers.

These six criteria synchronize well with other pertinent principals like the "The FAIR Guiding Principles for scientific data management and stewardship", which emphasizes automating the discovery of researchers' work through software algorithms by applying a succinct and measurable set of principles to make the work FAIR—Findable, Accessible, Interoperable, Reusable [214].

The availability of fundamental search elements in the search interfaces of both PubMed and GS is summarized in Figure 2.1, showing that PubMed's search interface fully implements the five recommended Boeker et al. search elements, while GS does not. However, GS's implementation of the forward citation search is much more popular than PubMed's implementation due to its heavy use of the *Cited by N* link.

The next sections contrast PubMed and GS for each of the search requirements. When the GS documentation describes how they support the search interface requirements, it is featured in text boxes. Screen shots were taken of all GS documentation featured in this commentary (S3).

2.4.1 Reproducibility of search results

Repeating search queries in PubMed always produced the same previous content in the results, plus the expected steadily rising hits resulting from the increasing coverage of the database over time. But when a query is run month to month in GS, numerous researchers have observed sudden jumps, both rising and falling, with large numbers of previous results lost [80] [213] [78].

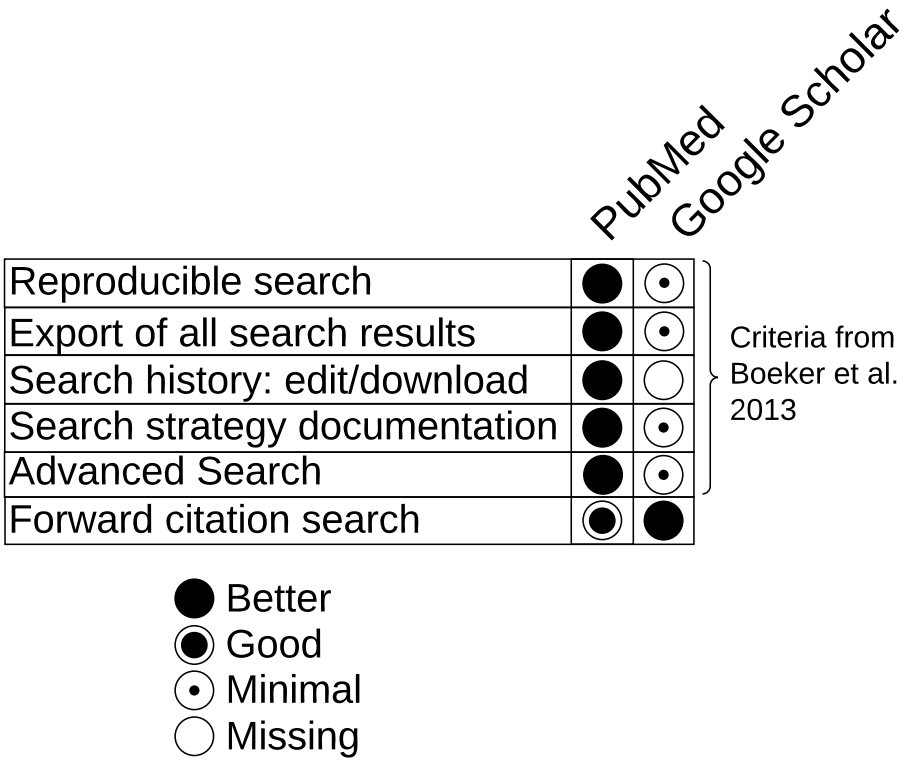

Figure 2.1: **Scientific search interface requirements.** PubMed fully implement's Boeker et al.'s required list of characteristics for systematic search interfaces, while GS's implementation provides minimal support. But PubMed does not implement the extremely popular *Cited by N* links seen throughout Google Scholar.

2.4.2 Search results can be exported in full

Search results in PubMed, even those not displayed on the screen, may be exported in full up to a maximum of 10,000 results. PubMed export formats include short summaries, files for import to citation software (like EndNote), PMIDs, abstracts, or a comma separated values (csv) file containing a list of data.

Searches in GS are limited to 1,000 results maximum and cannot be exported in bulk (Box 1), as described in their help documentation (S3 Figure 1):

Box 1. GS Search results

Can I see more than 1,000 search results?
Sorry, I can only show up to 1,000 results for any particular search query. Try a different query to get more results.

How do I get bulk access to records in Google Scholar?
Sorry, we're unable to provide bulk access.

Researchers who write a script to download GS search results programmatically, quickly discover the downloaded results are halted (Box 2) upon reaching an unspecified limit and then find this on the help page (S3 Figure 1):

Box 2. GS programmatic bulk export

I wrote a program to download lots of search results, but you blocked my computer from accessing Google Scholar. Can you raise the limit?

Err, no, please respect my robots.txt when you access Google Scholar using automated software. As the wearers of crawler's shoes and webmaster's hat, I cannot recommend adherence to web standards highly enough.

GS explicitly states in their documentation that they will return a maximum of 1,000 results for any search query (Box 1). The number of search results appears at the top of the list as "N result" in PubMed and "About N results" in GS. If "N" is less than 1,000 results in GS, researchers may think they can copy and paste all "N" results, 20 citations at a time, by clicking and

clicking and clicking, advancing slowly through the search results. But researchers may be surprised to find that even if "N" is less than 1,000 in GS, some of the "N" results may be missing [213].

PubMed can display 10, 20, 50, 100, or 200 results at a time on one page. GS can display 10 or 20 results. Clicking "Show more" in PubMed causes another set of results to be appended to the current results on the screen. The previous result sets are visible by scrolling up or pressing the browsers back button, which will cause the view to move to the previous divider and will not cause any results to disappear.

The "best match" relevance sort ordering in PubMed, described in a recent freely available peer-reviewed research article [61], uses a modern machine learning algorithm that is trained with aggregated user searches. The "best match" algorithm uses dozens of features to sort a list of citations, but its developers find that the most important document features are publication year and past usage. Additionally, recently published papers are given extra elevation in the sort list so that they will not be missed [61].

Users can find GS search algorithm components described in a variety of locations including a 1999 "technical report" on the website for the Stanford Digital Library Technologies Project which ended in 2004 [160], major updates as reported in a 2011 New York Times "Week in Review" piece [124], and numerous Google patents.

GS favors highly cited papers and ranks them at the top of the sort list [132] so recent papers are more likely to be many pages back, making them harder to find. It is important to understand GS's sort practices to be able to estimate which results over the 1,000 maximum were excluded.

2.4.3 Search history

PubMed records a history of every user search query and that user history is available as an interactive list where previous queries can be chained together and individual queries can be deleted to simplify the list. The full sequence of queries can be downloaded. GS has no similar search history.

2.4.4 Search strategy documentation

In addition to a comprehensive user guide, PubMed provides training in the form of tutorials, online training modules, quick tours, classes, and handouts. For further support, PubMed allows users to enter a question by clicking on the feedback link always shown at the bottom right corner of the page to bring up a contact form. A real person usually responds within the next business day or two. PubMed plans to move the feedback link to a "Contact Us" link located at the bottom of each web page now that "The New PubMed" is now the default link.

To access GS's contact form, click on questions like these (Box 3, S3 Figure 2 and 3):

Box 3. GS contact

I have noticed an error in a court opinion you are providing. What I can do to help fix it?

How do I remove a 'Cached' (or 'View as HTML') link from your search results?

2.4.5 Search string builder

The link to PubMed's advanced search is immediately below the main search query box, making access straightforward and efficient. The PubMed advanced search builder guides the user in building queries using more than 30 search fields, Boolean expressions (formed with AND, OR and NOT), and linking previous queries from the history. Additionally, users can customize the query entered in the query box.

The Boolean operators AND, OR, and NOT are required as it has been found that ranked retrieval alone, such as that found in GS, is not sufficient for a systematic search requiring high recall [101]. High recall ensures that all the expected matches appear in the search results [207]. These features give researchers the ability to fine-tune which results are included and which are not [87].

GS's advanced search only offers access to three fields, 'authored', 'published', and 'dated', compared with PubMed's 30 search fields. There is no support for full Boolean search [80], and no ability to string together previous queries. The link to the GS advanced search documentation is still described as being located immediately to the right of the main GS search box (S3 Figure 6). But the link has been moved away from the right of the main search box to under a menu icon on the upper left-hand corner of the GS web page (S3 Figure 7).

2.4.6 Forward Citation Search

PubMed has a forward citation search which can be accessed by opening the PubMed page for a single chosen article (S1 Figure 1). If the paper has citations, scrolling to the bottom of the page will show a "Cited by" section (S1 Figure 2, red 3) which lists the total number of citing papers in the section header and shows the first few papers in the section body. The full list of citing papers may be downloaded from PubMed in a variety of formats, including text or comma separated values (csv), by clicking the *See all cited by articles* link (S1 Figure 2, red 3a) and pressing the "Save" button (S1 Figure 5). But the web page showing the list of citing papers contains no citation count information for any articles on the page (S1 Figure 5). To see the citation count of each of the citing papers, the researcher must click on each citing paper one by one to open the individual paper's web page and scroll down to that paper's "Cited By" section, making choosing the next paper to explore a slow and laborious process (S1 Figure 1 and 2). I would like to see *Cited by N* links ubiquitously featured on all citations in a list (S1 Figure 6, red boxes). I rate the *Forward citation search* feature as "Good" rather than "Better" (Figure 2.1) because the *Cited by N* links do not appear (S1 Figure 5) in PubMed.

In GS, clicking the *Cited by N* link of a specific paper will open a web page with a list of papers citing the specific paper (S1 Figure 3 and 4). Each paper in the list has a *Cited by N* link (S1 Figure 4, red boxes), making it easier to compare the citing papers appearing in the list (S1 Figure 4, boxed in red). Unlike PubMed, there is no way to download the list of all citing papers in bulk. I rate this feature as "Better", even though it is not possible to compare

all search results in a single view, because of the usefulness and popularity of the GS *Cited by N* link.

PubMed is missing the *Cited by N* link on each paper in a list papers which is prominently featured in GS (S1 Figure 6), causing researchers to be lured towards GS and away from PubMed despite a grueling literature search experience in GS.

2.4.7 Scientific search feature summary

The advanced features recommended in 2013 [22] for an effective, exhaustive, and reproducible systematic review are fully implemented in PubMed, but was not implemented by GS in 2013, when Boeker did his study, and remains not implemented in 2020 [22] [80] [72].

And some GS features have made the search process more onerous. In 2008, GS search results could be displayed with 10-300 items per page [56]. Today, it is restricted to either 10 or 20 items per page (S3 Figure 12). Featuring search results at a maximum of 20 per page rather than 300 per page makes literature search more time-consuming, labor-intensive, and reduces a researcher's ability to visualize the search results as a whole. In 2013, Boeker concluded that GS was not ready as a searching tool for tasks where structured retrieval methodology is compulsory [22]. Almost a decade later, GS still can not be considered for such tasks.

2.5 Coverage of PubMed and Google Scholar

2.5.1 The coverage of PubMed

PubMed is a search interface and toolset used to access databases like MEDLINE and PubMed Central (PMC) as well as additional content like books and articles published before the 1960s. Over 30.5 million article records are accessible through the PubMed interface (Figure 2.2). The databases, MEDLINE and PMC, are separate entities whose combined articles comprise 94% of all of the coverage indexed by PubMed (S2). MEDLINE is a highly selective database started in the 1960s. PMC, started in 2000, is an open-access database for full-text papers that are free of cost to the reader.

2.5.2 The Coverage of Google Scholar

While the coverage of GS is not known, it is estimated to exceed all other currently available search systems since GS aims to index all of scholarly information that is electronically available [78]. This is a principal reason for its standard-setting citation index, which is used to replace "N" with a number in GS's forward citation search via *Cited by N*. The size and scope of GS remains unknown despite having been the subject of sizable research efforts since its creation [80] [132].

2.5.3 Journals covered

The GS documentation instructs researchers who want to know if a specific journal is covered to choose a "statistical sample" of articles published by the

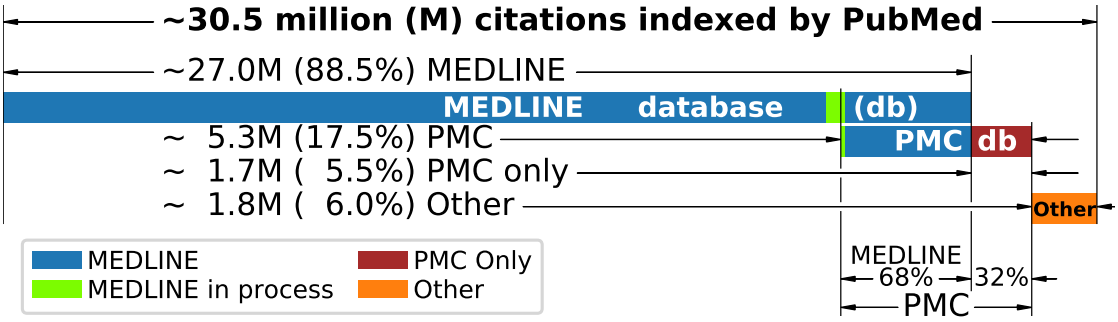

Figure 2.2: **Most of the coverage of PubMed is indexed in the MEDLINE database and the PMC database.** The coverage of PubMed is shown on the horizontal axis. The top two bars on the vertical axis show two overlapping databases indexed by PubMed. The bottom orange bar indicates PubMed citations not found in the two major databases. About 88.5% of the ~30.5 million citations accessible through PubMed are in the MEDLINE database (top blue bars) or are about to be added to (top green bars) the MEDLINE database (top blue and green bars). The MEDLINE papers that are free full-text and are also indexed in the PMC database (middle blue and green bars) comprise over 68% of papers indexed in PMC. About 5.5% of PubMed papers are only available in PMC (middle brown bar). Almost all of the remaining 6% of full-text papers (bottom orange bar) are behind a paywall. I queried for and downloaded [186] PubMed count data and created the figure with a script available in *pmidcite*.

journal and search for each paper using its title in the search box (Box 4, S3 Figure 4):

Box 4. GS journal coverage

Which specific journals do you cover?

Ahem, I index papers, not journals. You should also ask about my coverage of universities, research groups, proteins, seminal breakthroughs, and other dimensions that are of interest to users. All such questions are best answered by searching for a statistical sample of papers that has the property of interest - journal, author, protein, etc. Many coverage comparisons are available if you search for [allintitle:"google scholar"], but some of them are more statistically valid than others.

In contrast to the GS approach, researchers can download PubMed's complete list of journals currently indexed in MEDLINE and deposited in PMC by following the *journals* link found on PubMed's home page. If a journal is on MEDLINE's approved journals list, papers are automatically indexed by PubMed (S2).

2.5.4 Indexing procedure for individual manuscripts

If an individual author manuscript was accepted into a journal that is not on MEDLINE's approved journal list, the requirements to submit the manuscript for deposit into PMC and indexing by PubMed are as follows. The work must be funded by an approved agency, peer-reviewed, accepted into a journal, and free to access electronically. If these criteria are met, the paper may be submitted to the NIH Manuscript Submission system (NIHMS) for potential indexing in PMC after the paper has been successfully vetted in NIHMS.

In GS, the requirements are the article must be contained in a pdf file whose contents include a title, list of authors, and bibliography and uploaded to a website. The affect of GS's regularly crawled data and loose indexing

policies is that GS indexes records that are non-academic. For example, the GS policies for 'author manuscripts' has resulted in a number of lunch menus that are stored as a pdf file online to be indexed as scholarly citations in GS with various food items being listed as authors (S3 Figure 10 and 11).

Additionally, some researchers have demonstrated that it is possible to deliberately trick the crawler and inflate the GS citation score [125].

2.6 Forward citation search

The NIH Open Citation Collection (NIH-OCC) [92], is a free public citation database, which liberates researchers from the constraints of citation data that were previously locked behind barriers, such as citation lists which were not downloadable in bulk. Having full access to citation data could allow researchers to perform more efficient literature searches and analyze publishing trends in biomedicine.

The NIH citation database differs from GS's citation database in coverage, usability, and content. The coverage and content of GS is huge and covers many disciplines, while the coverage of the NIH citation database is limited currently to about 30.5 million manuscripts that were assigned a PubMed ID (PMID). The usability of iCite citations is extremely high because they are accessible for free through the NIH "iCite" web site and downloadable in bulk through the NIH Application Programming Interface (API). Citations can not be downloaded in bulk in GS.

The citation counts in GS will be higher because their index is massive. I have not experienced that the differing citation counts when using *pmidcite*

versus GS are a hindrance during exploratory search tasks because the data needed to decide the next paper to investigate is how one paper performs relative to another. If all citation counts are scaled down in NIH's "iCite" compared to GS, I still can successfully compare the performance of papers relative to one another.

Additionally, in NIH's "iCite", new papers are easy to find and compare, even if they have few citations. Having the data, even if it is scaled down compared to GS, to choose the next paper will speed the exploratory literature search faster than having all of the citations that are available in GS, but not available in other search systems. Once the researcher has become familiar with the subject through their exploratory literature search, then they may choose to use GS to see what might have been missed.

I have tested the practical usage of a *Cited by N* link by creating a set of command-line interface (CLI) scripts and a Python library, called *pmidcite*, which glue PubMed search results and NIH's "iCite" citation data together using PMIDs to provide functionality that is equivalent to having the *Cited by N* link. The results were so successful that I hope PubMed can expand the access to all biomedical researchers, even if they do not use a CLI by adding the links, *Cited by N* and *N References* (S1 Figure 6), to the PubMed GUI as soon as possible.

2.6.1 NIH's freely available citation data

PubMed does not have a clickable *Cited by N* link for all the citations, making it difficult to choose the next paper to investigate (S1 Figure 5 and 6). But equivalent functionality can be had if, for a selected paper, the researcher

downloads from PubMed the full list of citing papers as a list of PMIDs (S1 Figure 7) and uploads this PMID list of citations to iCite for analysis (S1 Figure 8). The list can then be sorted in the "citations" tab (S1 Figure 9, red 2) by *Total Citations* by clicking on the *Total Citations* column header under the *OpenCites* tab (S1 Figure 9, red 3).

But comparing papers only using number of citations is problematic because papers in small niche fields may get considerably less citations than papers in large fields; both papers may be of relatively equal scientific influence in their respective communities. The NIH normalizes the number of citations that a paper receives by comparing it to the citation numbers of papers in its co-citation networks. This measurement is called the Relative Citation Ratio (RCR) [93] and can be used to sort a list of PMIDs. Only citation count is offered by GS and it can't be used for sorting search results.

2.6.2 The open-source software of *pmidcite*

Functionality equivalent to having a *Cited by N* link can be had from the command-line shown in the following example using a selected paper with a PMID of 25505874 by typing "`icite 25505874 --verbose`" and pressing the "Enter" button. This causes citation counts to be downloaded from NIH's iCite and a report to be written to the screen or to a file. In the report, the number of citations is seen under the "`cit`" column for the user-requested paper ("`TOP`"), the full list of it's citing papers ("`CIT`") and references ("`REF`"):

```
$ icite 25505874 --verbose

typ PMID    year   cit au_cnt(1st author) title of article
--- -------- ---- ---- ------------------------------
TOP 25505874 2014   10 au[02](B Ian Hutchins) Capture of microtubule plus-ends ...
CIT 27014940 2016   12 au[11](B Ian Hutchins) CCDC141 Mutation Identified in A ...
CIT 30902847 2019    6 au[09](Daisuke Inoue) Actin filaments regulate microtub ...
CIT 31355196 2019    0 au[04](Hyun-Ju Cho) Nasal Placode Development, GnRH Neu ...
CIT 26613184 2016    0 au[05](Lulu Huang) Laser Activated Electron Tunneling B ...
REF 12600310 2003 2522 au[02](Thomas D Pollard) Cellular motility driven by as ...
REF 10899992 2000  623 au[05](T A Klar) Fluorescence microscopy with diffracti ...
```

The NIH values based on a paper's RCR are also available in *pmidcite*, but are not shown here. For more information regarding *pmidcite* and to see options for sorting citing papers from the CLI, see S1 and `https://github.com/dvklopfenstein/pmidcite`.

2.7 Details regarding *pmidcite*

First, I show the *Cited by N* links as they currently appear in PubMed and Google Scholar (GS). Second, I show the placement of the *Cited by N* links and the *N References* links that I would like to see in the PubMed Graphical User Interface (GUI). Finally, I demonstrate a new method to annotate PubMed search results with citation data from the NIH Open Citation Collection (NIH-OCC) for command-line interface (CLI) users.

2.8 The current PubMed forward citation search

Figures 2.3 and 2.4 show how to find both the citation count (Figure 2.4 red 3) and the reference count (Figure 2.4 red 4) in PubMed by searching for a paper using its title. Figure 2.5 shows how to find the citation count (Figure 2.5 red 2), but not the reference count, in Google Scholar (GS).

Clicking on the *See all "Cited by" articles* link (Figure 2.4, red 3a) in PubMed will open a new page showing the list of citations. This is a forward citation search. Clicking on the *Show all 35 references* link (bottom of Figure 2.4) will open a new page showing the 35 references. This a reverse citation search. Both forward and reverse citation search functionality are useful in exploratory searches and systematic searches. The reference count and reverse search functionality is not available in GS.

GS shows a total of 353 citations (Figure 2.5, bottom left red box) versus PubMed's 68 citation count. Since GS indexes preprints, white papers, theses, and papers not related to chemical and health sciences, it will show a higher citation count than PubMed. Higher citations counts do not always reflect better papers. For example, important research may have lower citation counts if it was published in a less prestigious journal. Also, citation count alone does not reflect how a paper is performing next to its peers. For example, a mathematical paper with a small number of citations may have a greater effect in its field than an oncology study having hundreds of citations [145].

2.9 Google Scholar's popular *Cited by N* link

Clicking on the *Cited by 353* (Figure 2.5, bottom red box) in GS will open a page showing the first 10 or 20 citations (Figure 2.6). Researchers can not download the full 353 citations in bulk and instead must click and click and click to page through citation results 10 or 20 citations per page. However, GS's *Cited by N* links (Figure 2.6, red boxes) are exceedingly popular and are missing in

PubMed. Citation count data can aid the researcher in deciding which paper to examine next.

Clicking on *See all "Cited by" articles* (Figure 2.4 red 3a text) link in PubMed will open a new page showing the list of citations (Figure 2.7). Researchers can download the full 68 citations, plus the original paper seen at the top, in a variety of formats including comma-separated values (csv), PubMed ID (PMID) lists, and summaries using the "SAVE" button boxed in red (Figure 2.7).

But the *Cited by N* link is missing. The additional links I would like to see in the PubMed interface are drawn in (Figure 2.8 red boxes). Note that only the top paper has "Cited by 108" because the following 3 papers are new and do not yet have citations. In addition to the *Cited by N* link for easier forward search citation, I would like to see an *N References* link to aid in reverse citation search.

Notice that PubMed's "Best Match" relevance sorting shows new papers at the top of the citation list, while GS's relevance sorting shows highly cited papers at the top. GS favoring highly cited papers by ranking them at the top of their list [132], coupled with the laborious work of clicking and clicking to traversing the citation list can cause highly cited papers to continue to gain more citations, while important new papers may be overlooked.

2.10 Attach citation data to PubMed results

After finding an article in PubMed, to obtain the equivalent functionality of having ubiquitously featured *Cited by N* links and *N References* links (Figure 2.8

red boxes) scroll down to the section showing all "Cited by" articles (bottom of Figure 2.3 and top of 2.4) and save all results as PMIDs to a file (Figure 2.9).

Next, the file can be annotated with citation and translational data from the NIH Open Citation Collection, iCite, using either the graphical user interface (GUI) method or the command-line interface (CLI) method described below. These methods provide information equivalent to that found using GS's *Cited by N*. In this example, PubMed writes the list of PMIDs into a file named "pmid-26379270-set.txt."

2.10.1 Using the Graphical User Interface (GUI)

If you are not a command-line interface (CLI) user, or you are a CLI user and want to see the iCite graphs generated for your data, go to the iCite website at `https://icite.od.nih.gov/analysis` and click on the "New Analysis" link at the top of the page. Load the file, in this case "pmid-26379270-set.txt", containing the your PMIDs (Figure 2.10) and press the "Process" bar. Click on the "Citations" tab (Figure 2.11 red 2), which is next to the tabs, "Influence" and "Translation." You will then see a list of the citing PMIDs, each annotated with citation count (Figure 2.11, red 3), reference count, and the Relative Citation Ratio (RCR), which is a NIH metric that compares the performance of a paper against its peers in the same citation network [93]. An RCR is a normalized metric of influence that has a transparent methodology and is free [93] [145]. For example, a RCR of ˜12 means that a paper is being cited 12 times as expected as compared to its peers.

2.10.2 Using *pmidcite* from the command-line interface (CLI)

A command-line interface (CLI) can be preferable to a Graphical User Interface (GUI) because: processing can be automated from a script, time-consuming mouse clicking is reduced, and more data can be seen at once on a text screen than in a browser, giving the researcher a better overall impression of the full set of information.

Linux and Mac users already work from the command-line. Windows users can get that Linux-like command-line feeling while still running native Windows programs by downloading Cygwin from
`https://www.cygwin.com/`

Run icite

The open-source software, *pmidcite*, is a Python package written by this first author and is freely available at
`https://github.com/dvklopfenstein/pmidcite`. NIH's iCite has a simple, clean, fast Application Programming Interface (API) which allows their citation data to be downloaded over the internet by libraries like *pmidcite*. NIH's iCite returns a succinct record for each PMID containing citation, reference counts and more. When you run *pmidcite*, you are downloading the same data as seen in the previous GUI example. To run the "icite" script, which is part of the *pmidcite* library, do:

```
$ icite -i pmid-26379270-set.txt -o icite-26379270-set.txt
    69 WROTE: icite-26379270-set.txt
```

Running "`icite`" causes the PMIDs to be read from the PMID input ("`-i`") file specified with the argument "`-i pmid-26379270-set.txt`."

The PMIDs are sent to NIH's iCite and citation data is downloaded and annotated to the PMIDs. The annotated PMID report is written to the output ("`-o`") file specified with the argument "`-o icite-26379270-set.txt`." The "`69 WROTE: icite-26379270-set.txt`" is text printed by icite letting the user know that 69 PMIDs are annotated with citation data and written to "`icite-26379270-set.txt`." The PMIDs in the file are printed in the same order that they were sorted in PubMed with the "Best Match" algorithm before being downloaded.

View citation data

The citation data is formatted in columns separated by spaces. To print the column headers and the citation data, two commands are run back-to-back. The first command is "`icite -H`", which causes the column headers to be printed. The semi-colon ("`;`") signifies the end first command. The second command is "`grep TOP icite-26379270-set.txt`." The command, "`grep`", is like the "find" command in word-processing programs, but it is used for ASCII text files. In this example, I are finding and printing all lines containing the text, "`TOP`", in the "`icite-26379270-set.txt`" citation report. The text, "`TOP`", signifies that the line contains citation data for a user-requested PMID, rather than a PMID that is a reference or citing paper of a user-requested PMID.

```
$ icite -H; grep TOP icite-26379270-set.txt
TYP PMID     RP HAMCc   % SD YR   cit cli ref au[00](authors) title
TOP 26379270 R. H....  99 4 2015   108   0  12 au[04](Neal Robert Haddaway) The ...
TOP 32454855 .. H....  -1 i 2020     0   0  23 au[07](Huimin Jin) Effectiveness ...
TOP 32408896 R. H....  -1 i 2020     0   0  79 au[03](Emmanuel N-B Quarshie) Se ...
TOP 32235580 .. H....  -1 i 2020     0   0  40 au[02](Charlene Elliott) The Pow ...
```

Column headers

The column header line, which starts with "`TYP PMID ...`" is printed first by running "`icite -H`." Under the column header line is a list of papers, starting with the chosen Haddaway paper, and followed by papers that cite it. The columns, "`cit`" and "`ref`", contain the citation count and reference count for the paper on the same line. There is a high citation count of 108 for the Haddaway paper. I only show 3 of the 108 citing papers for brevity. The 2020 papers have a citation count of zero, likely due to being so recently published.

Notice that the citation counts for PubMed (68), NIH's iCite (108) and for GS (353) do not agree. The decision of which citation counts to use in the recommended new PubMed *Cited by N* links should be decided by PubMed. I do not believe that the differing citation counts are a major hindrance for a researcher in choosing their next paper to investigate while doing exploratory search. The data needed to choose the next paper to investigate is how new a paper is performing against its peers. Having the citations scaled lower, as in the PubMed citation counts will likely not have a great effect on displaying how each paper performs relative to others. Having the data to choose the next paper will speed the exploratory literature search faster than having all 353 GS citations. Once the researcher has become familiar with the subject through their exploratory literature search, then they can search in GS for additional papers.

The columns "`PMID`" and "`YR`" (YEAR) are the PubMed ID and the year that the paper was published.

Column header key

To better understand the remaining columns containing additional NIH iCite data, print the key as shown below ("`icite -k`"). The key indicates that first the three letters of the line describes a paper's "type" (`TYP`), which include user-requested paper (`TOP`), citing paper (`CIT`), or reference (REF). The keyword makes grepping easier. Doing "`grep TOP [file.txt]`", will display one descriptive line for each user-requested PMID.

```
$ icite -k

KEYS TO PAPER LINE:
    TYP PubMedID RP HAMCc % nihSD YEAR x y z au[A](First Author) Title of paper

TYPe of relationship to the user-requested paper (TYP):
    TOP: A user-requested paper
    CIT: A paper that cited TOP
    CLI: A clinical paper that cited TOP
    REF: A paper referenced in the TOP paper's bibliography

NIH iCite details:

  PubMedID: PubMed ID (PMID)

     RP section:
     -----------------------------------
         R: Is a research article
         P: iCite has calculated an initial Relative Citation Ratio (RCR) for  ...

     HAMCc section:
     -----------------------------------
         H: Has MeSH terms in the human category
         A: Has MeSH terms in the animal category
         M: Has MeSH terms in the molecular/cellular biology category
         C: Is a clinical trial, study, or guideline
         c: Is cited by a clinical trial, study, or guideline

     NIH section, based on Relative Citation Ratio (RCR):
     -----------------------------------
         %: NIH citation percentile rounded to an integer. -1 means "not deter ...
     nihSD: NIH citation percentile group: 0=-3SD 1=-2SD 2=+/-1SD 3=+2SD 4=+3S ...

     YEAR/citations/references section:
     -----------------------------------
      YEAR: The year the article was published
         x: Number of unique articles that have cited the paper
         y: Number of unique clinical articles that have cited the paper
         z: Number of references
     au[A]: A is the number of authors
```

The sections, "RP" and "HAMCc" contain descriptive citation and translation data from the NIH-OCC and are explained in the key. The "y" number near the bottom of the key is the number of clinical trials that have cited the paper. If any clinical trials cite the paper, a "c", rather than a "." is printed in the "c" position of the "HAMCc" section.

The NIH section

The key to the NIH section is:

```
NIH section, based on Relative Citation Ratio (RCR):
-----------------------------------
    %: NIH citation percentile rounded to an integer. -1 means "not deter ...
nihSD: NIH citation percentile group: 0=-3SD 1=-2SD 2=+/-1SD 3=+2SD 4=+3S ...
```

The RCR is not used in the CLI display because the NIH developers calculated the NIH citation percentile using the RCR, making the NIH citation percentile a reasonable proxy for the RCR score. The NIH citation percentile is better suited to the CLI because it takes less space than the RCR and is better for sorting on the command-line.

If the NIH percentile was displayed and used for sorting papers, the result would be unacceptably rigid sorting. Sorting such that a paper with a 55% NIH percentile is always shown ahead of a paper with a 45% percentile is not helpful, since both papers have a good citation performance. I created and added "NIH percentile group" ("nihSD"), so papers could be sorted by large performance groups, like good, high, and very high (Figure 2.12).

The performance groups are numbered from 0 to 4, with 0 being the lowest performing papers and 4 being the highest performing papers. Recent papers

that have not had enough time to accumulate an RCR or NIH percentile are given an "i" as a placeholder for some number to be determined later.

Summarizing a single paper

To print the summary for a single paper, do:

```
$ icite 27846867 -H
TYP PMID     RP HAMCc   % SD YR   cit cli ref au[00](authors) title
TOP 27846867 .. H....  82 2 2016    13  0  17 au[03](Claire Stansfield) Explor ...
```

This paper has 13 citations ("`cit`") and 17 references ("`ref`"). Its NIH standard deviation group ("`SD`") is a 2, so it is a paper that is performing well with its peers. With an NIH percentile ("`%`") of 82%, it is almost in the high performing papers. The "`PMID`" for the paper is "`27846867`." The "`-H`" argument causes the column headers to be printed.

Examining the citations of a single paper

To get all citation details for a single paper, do:

```
$ icite 27846867 --verbose --no_references
TOP 27846867 .. H....  78 2 2016    16  0  17 au[03](Claire Stansfield) Explor ...
CIT 32511888 R. .....  -1 i 2020     0  0  32 au[03](Simon Briscoe) How do Coc ...
CIT 31343759 .. H....  -1 i 2020     0  0  12 au[03](Louise Harriss) Building  ...
CIT 31541534 .. H....  -1 i 2019     3  0  80 au[04](Anthea Sutton) Meeting th ...
CIT 30993756 R. H....  -1 i 2019     2  0  13 au[11](Stefanie Buckner) Dementi ...
CIT 29353363 RP H....  -1 i 2019     2  0   6 au[01](Marko Curkovic) The Impli ...
CIT 31866923 .. H....  -1 i 2019     0  0  51 au[07](Meg E Morris) Boxing for  ...
CIT 29783954 RP H....  98 4 2018    26  0 101 au[07](Monika Mueller) Methods t ...
CIT 29193834 RP H....  90 3 2018     8  0  29 au[05](Chris Cooper) Supplementa ...
CIT 29179733 .. H....  90 3 2017    10  0  49 au[04](Chris Cooper) A compariso ...
CIT 29065246 .P H....  77 2 2018     7  0  51 au[01](Simon Briscoe) A review o ...
CIT 30177007 RP H...c  66 2 2018     5  2  40 au[05](Filippo Bianchi) Restruct ...
CIT 30424741 RP H....  66 2 2018     5  0  16 au[02](Marko Ćurković) Bubble ef ...
CIT 30340498 RP HA..c  42 2 2018     3  2  44 au[05](Filippo Bianchi) Interven ...
CIT 30453942 RP H....  17 2 2018     1  0  58 au[05](N Mahmoodi) Are publicly  ...
```

The "`--verbose`" option causes the citing papers to be printed.

The "`--no_references`" option prevents the references from being printed.

The requested paper, PMID "`27846867`" appearing on the "`TOP`" line, is printed along with its citing papers. Because the citations were downloaded using NIH-OCC and not from PubMed, PubMed's "Best Match" relevance sort has not been run.

So *pmidcite*'s default sort of the citations ("`CIT`") begins with sorting the group numbers in this order: "`i, 4, 3, 2, 0`." Second, it sorts by year so that the newest papers are at the top of each group. Third, it sorts by NIH percentile ("`%`").

Since all the newest papers do not have a NIH percentile yet ("`-1`"), the NIH percentile sort has no effect. For all new papers in the same year, the next sort uses the sum of the citations from both clinical ("`cli`") and non-clinical ("`cit`") papers. Finally, for new papers published in the same year that have the same number of citations, the final sort is done using the number of references ("`ref`").

To get all citation details for the references for a single paper do:

```
$ icite 27846867 --verbose | grep REF
REF 26379270 R. H....  99 4 2015   119  0  12 au[04](Neal Robert Haddaway) The ...
REF 26052848 R. .....  98 4 2014    86  0   8 au[03](Quenby Mahood) Searching  ...
REF 27686611 R. H....  94 3 2016    24  0  37 au[07](Jean Adams) Searching and ...
REF 26932789 R. H....  91 3 2016    29  0  11 au[03](Wichor M Bramer) Comparin ...
REF 26494010 R. H....  97 3 2015    49  0  13 au[05](Katelyn Godin) Applying s ...
REF 24360284 R. H....  86 3 2013    39  0  24 au[04](Wichor M Bramer) The comp ...
REF 25928625 R. .....  61 2 2015    14  0  13 au[04](Sandy Oliver) Capacity fo ...
REF 25889619 R. H....  61 2 2015    10  0   9 au[01](Simon Briscoe) Web search ...
REF 25031558 R. H....  76 2 2014    29  0  23 au[04](Julie M Glanville) Search ...
REF 24785398 .. H...c  66 2 2014    18  1  45 au[05](Rebecca W Rees) 'It's on  ...
REF 26052650 R. .....  51 2 2014    11  0  11 au[05](Tamara Rader) Methods for ...
REF 26052653 R. .....  44 2 2014     9  0  13 au[03](Claire Stansfield) Search ...
REF 24160679 R. H....  80 2 2013    31  0  40 au[03](Martin Boeker) Google Sch ...
REF 21439062 .. H...c  79 2 2011    35  2  24 au[04](Rebecca Rees) The views o ...
REF 11706930 R. .....  40 2 2001    18  0   0 au[03](G Eysenbach) Evaluation o ...
REF 23738438 R. H....  13 1 2013     3  0   0 au[01](Karen Blakeman) Finding r ...
REF 26061784 R. H....   8 1 2011     2  0  19 au[02](Karen Schucan Bird) Syste ...
```

The "`grep REF`" command causes only the references to be shown on the screen, not the user-requested ("`TOP`") paper or its citing papers ("`CIT`"). The default sort order of the references is the same as the default sort order for the citations.

To sort the most recent papers by reference count ("`ref`"), do:

```
$ icite 27846867 --verbose | grep -w i | sort -k10,10 -r
CIT 31541534 .. H....  -1 i 2019     3  0  80 au[04](Anthea Sutton) Meeting th ...
CIT 28789703 .. H....  -1 i 2017     0  0  36 au[02](Rosie Hanneke) Informatio ...
CIT 30993756 R. H....  -1 i 2019     2  0  13 au[11](Stefanie Buckner) Dementi ...
CIT 29353363 RP H....  -1 i 2019     2  0   6 au[01](Marko Curkovic) The Impli ...
```

To get only the most recent papers that have not yet been rated, "`grep -w i`" command looks for the lone "` i `", in the report lines, which represents the NIH SD data. The "`-w`" option on grep means "`i`" must be a "word" by itself, not embedded in another word. Another grep that would produce the same result is "`grep "-1 i"`."

This list is sorted by the number of references. So the paper with the most references ("`80`") is at the top and the paper with the smallest number of references is at the bottom ("`6`"). The "`-k10,10`" sort option causes the sort to only use the 10th column which is the reference count ("`ref`"). If you used "`-k10`", the sort would start from the text at the 10th column and ending at the end of the line. In this case, since all of the reference counts are different there is no difference in the sort when using "`-k10`" or "`-k10,10`". Because "`-k10`" is shorter to type, it may preferable to use in this circumstance.

Retaining a history

To retain a history of papers of interest, re-run "`icite`", appending the paper's NIH iCite information to a log file ("`-a lit.txt`"). If the name of the log file is "`lit.txt`", do:

```
$ icite 27846867 -a lit.txt
```

The "`lit.txt`" can be saved and revision-managed using a tool like git and can be a reminder as to which papers were of interest and the order they were found.

2.11 A comment on the *N References* link

Researchers can access a document's references using PubMed, but not GS. Only being able to traverse the literature in the forward direction by traversing to new papers that cited a paper, but not being able traverse backwards to references in the paper hinders a researcher's ability to dig deep into the literature.

Worse, it prevents a single author from confirming that a paper is erroneously marked as *Cited by* by another paper in GS. This potential breach in literature connectivity in GS may result in incorrect citations and the inability to measure the extent of this potential problem.

2.12 Conclusion

I hope to raise awareness that there are various types of search, including lookup tasks, exploratory search, and systematic search. Each search type

requires unique search system features. GS is the system used as the starting point of most searches, rather than specialized tools like PubMed, by most researchers due to its intuitive, accessible interface, fast response, and best in class coverage [149]. GS excels for simple lookup tasks, like finding a paper by entering its title in the query box [72].

Both GS and PubMed can be used for exploratory searches, but I urge biomedical researchers to use PubMed rather than GS, because PubMed is one of the top recommended primary sources for literature searches of peer-reviewed research in the biomedical sciences and has search feature criteria that GS has lacked since its inception. Command-line interface (CLI) users, especially, should consider using PubMed with search results annotated using NIH's "iCite" citation data because this functionality is available immediately through *pmidcite*.

Searching using the PubMed interface is a satisfying experience, even without the addition of the *Cited by N* link. But I hope that PubMed will soon add a clickable citation count link to every document entry in the search results list and to each paper listed in the document page sections, *similar articles*, *cited by*, *references*, and *suggested reading* so PubMed Graphical User Interface (GUI) users can enjoy similar benefits as CLI users.

GS fails to implement the required search criteria for systematic searches [22] and should not be used as a primary search tool for systematic reviews [22]. However, GS can be used as a secondary source [80].

Finally, I urge researchers to read the Gusenbauer and Haddaway paper to see how their own specialized search tool is or can be evaluated among the 28 extensively used academic search systems in the Gusenbauer study.

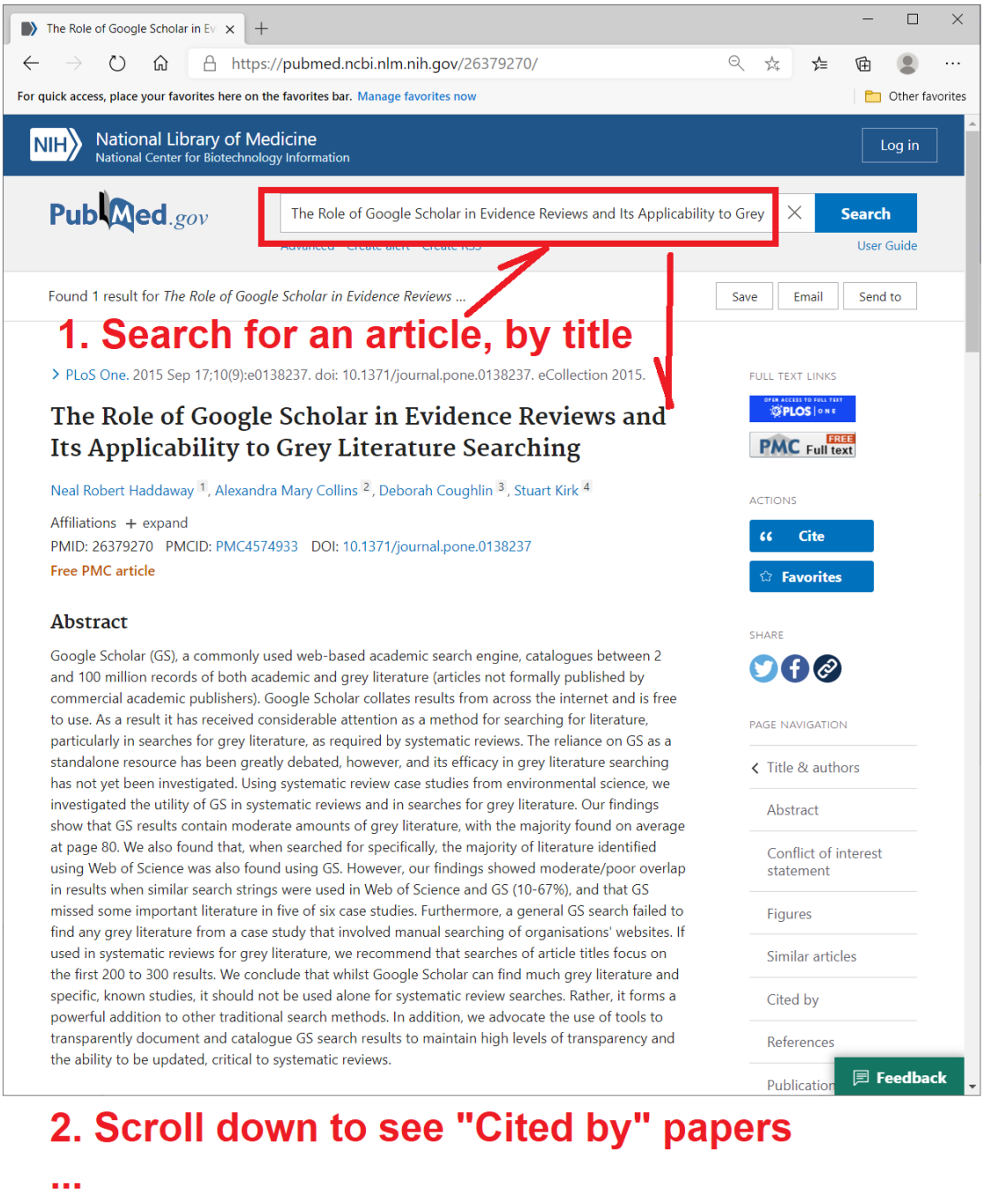

Figure 2.3: **Searching for a specific paper by it's title in PubMed.** First, enter the article's title in the search box. Then press enter. Scroll down to see the forward citation.

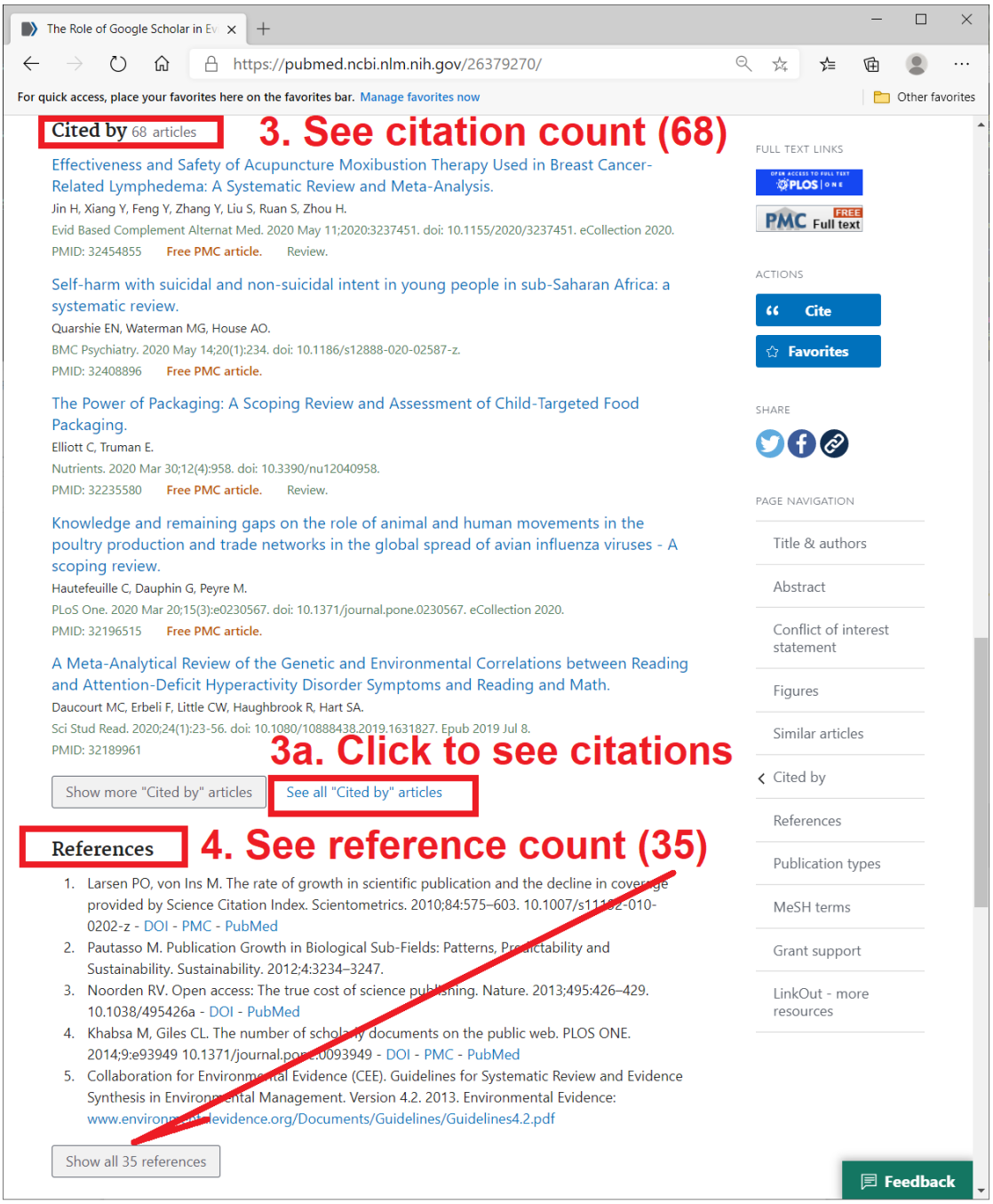

Figure 2.4: **Searching for a specific paper by it's title in PubMed.** After entering the article's title in the search box and pressing enter, scroll down to see the forward citation in the "Cited by" section. The header of the "Cited by" section and the "References" section contain the count of citing items. To open a page with the list of citations, click *See all "Cited by" articles*, marked as step 3a.

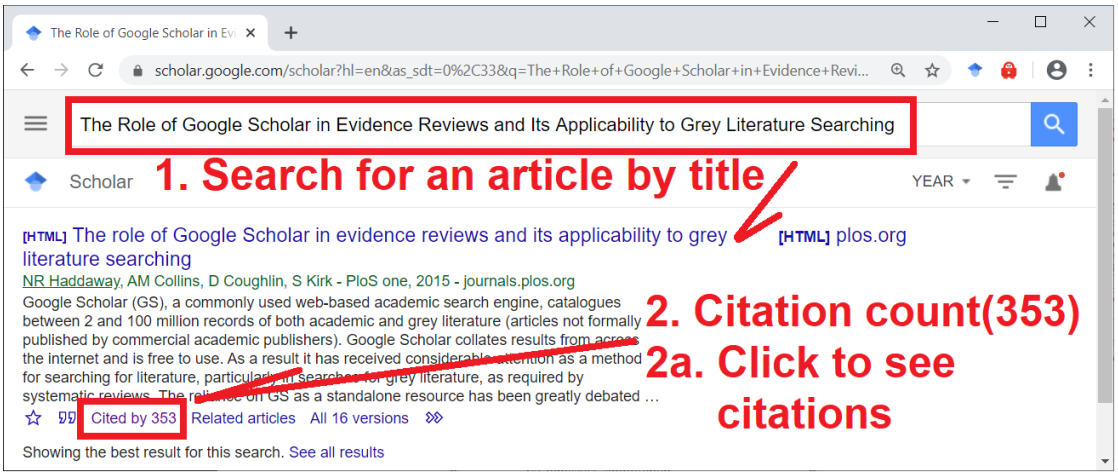

Figure 2.5: **Searching for a specific paper by it's title shows 353 citations in Google Scholar.** First, enter the article's title in the search box. Then press enter. Then see the citation count.

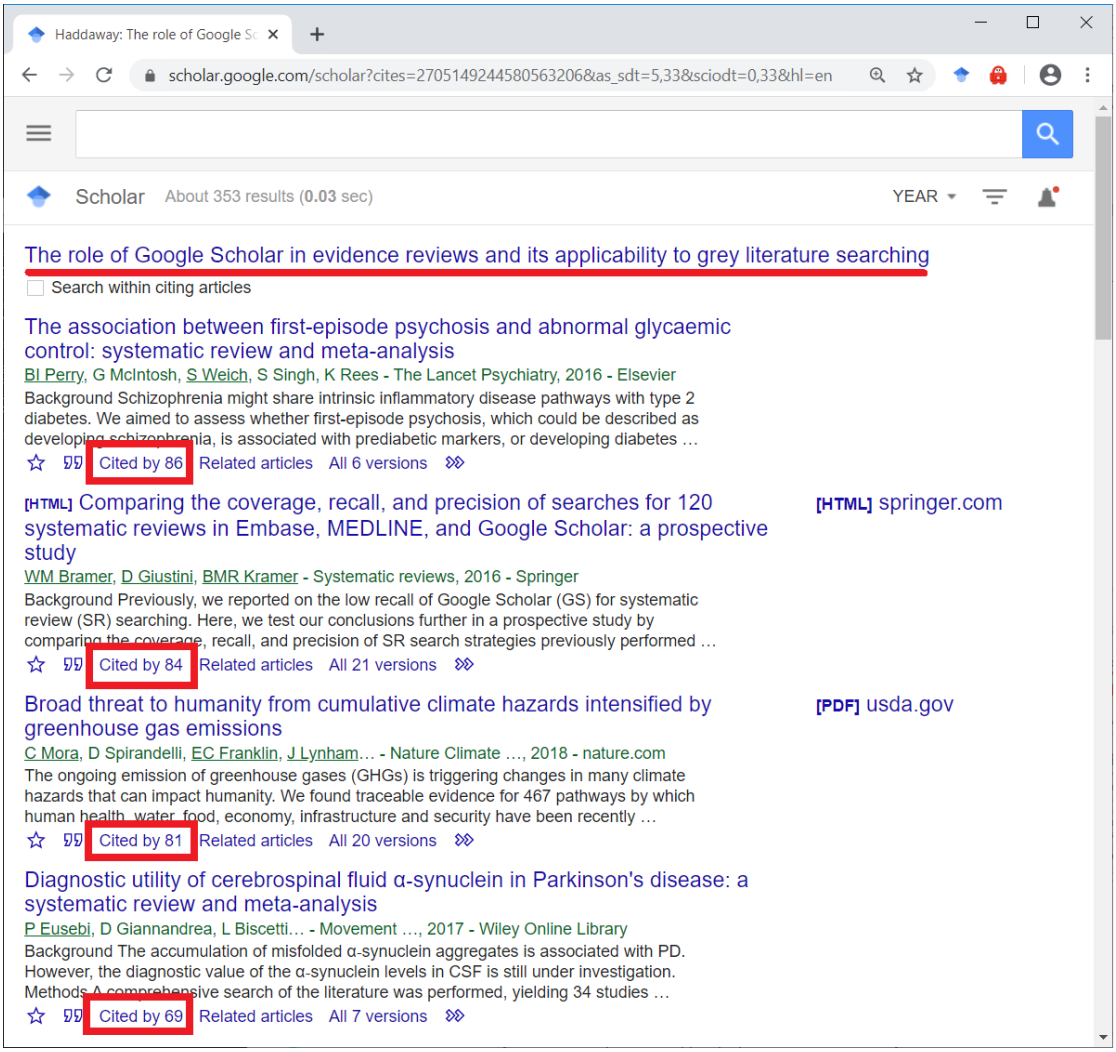

Figure 2.6: **The first four of 353 citations for a specific paper.** Each citation for the paper, "The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching" has its own citation count, shown in the red boxes, aiding the user to choose the next paper to investigate.

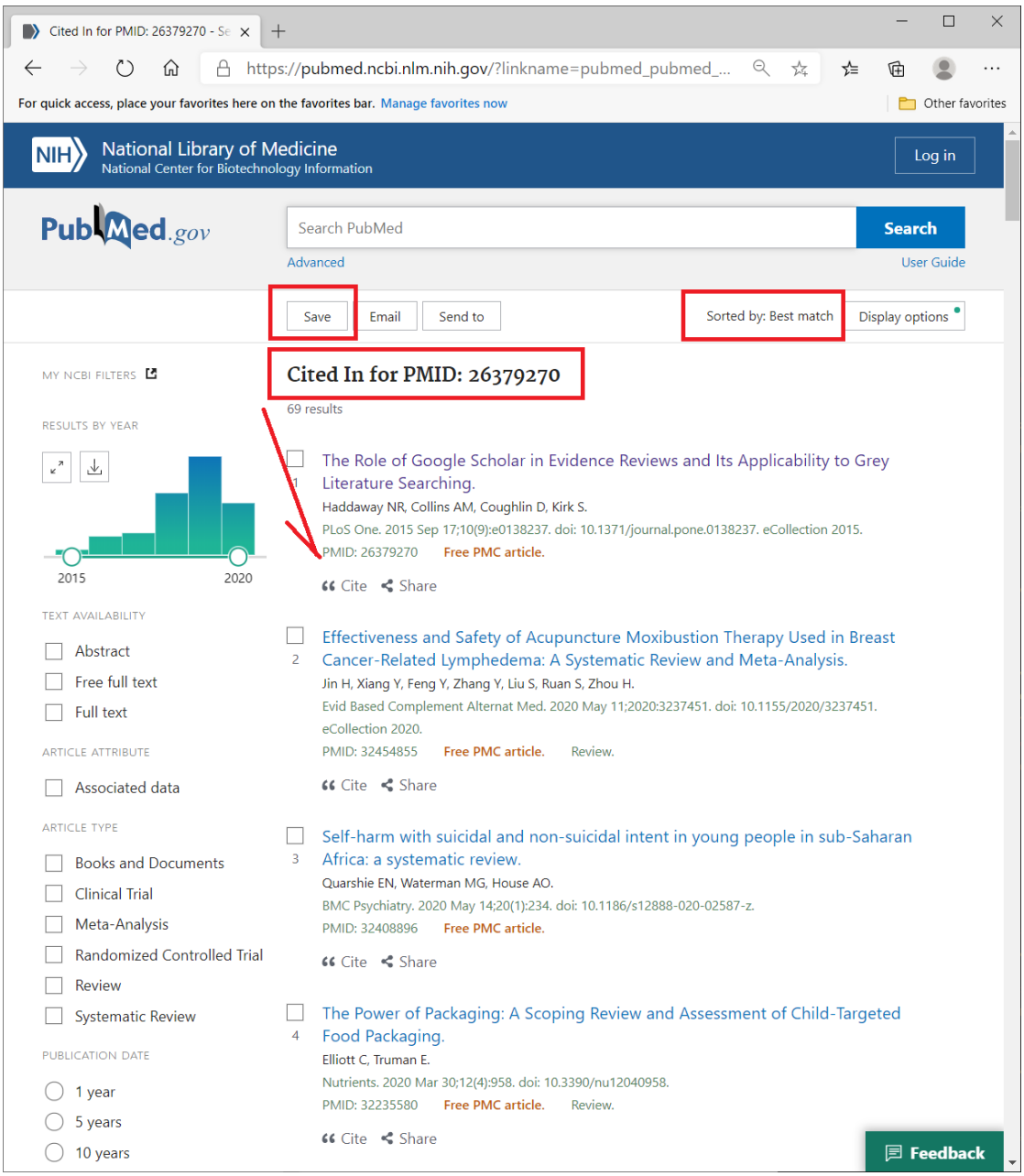

Figure 2.7: **The first three of 68 citations for a specific paper as they currently appear in PubMed.** The original paper, "The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching", is at the top of the list. All 68 citing papers can be downloaded or shown on the page by scrolling down and clicking, "Show more."

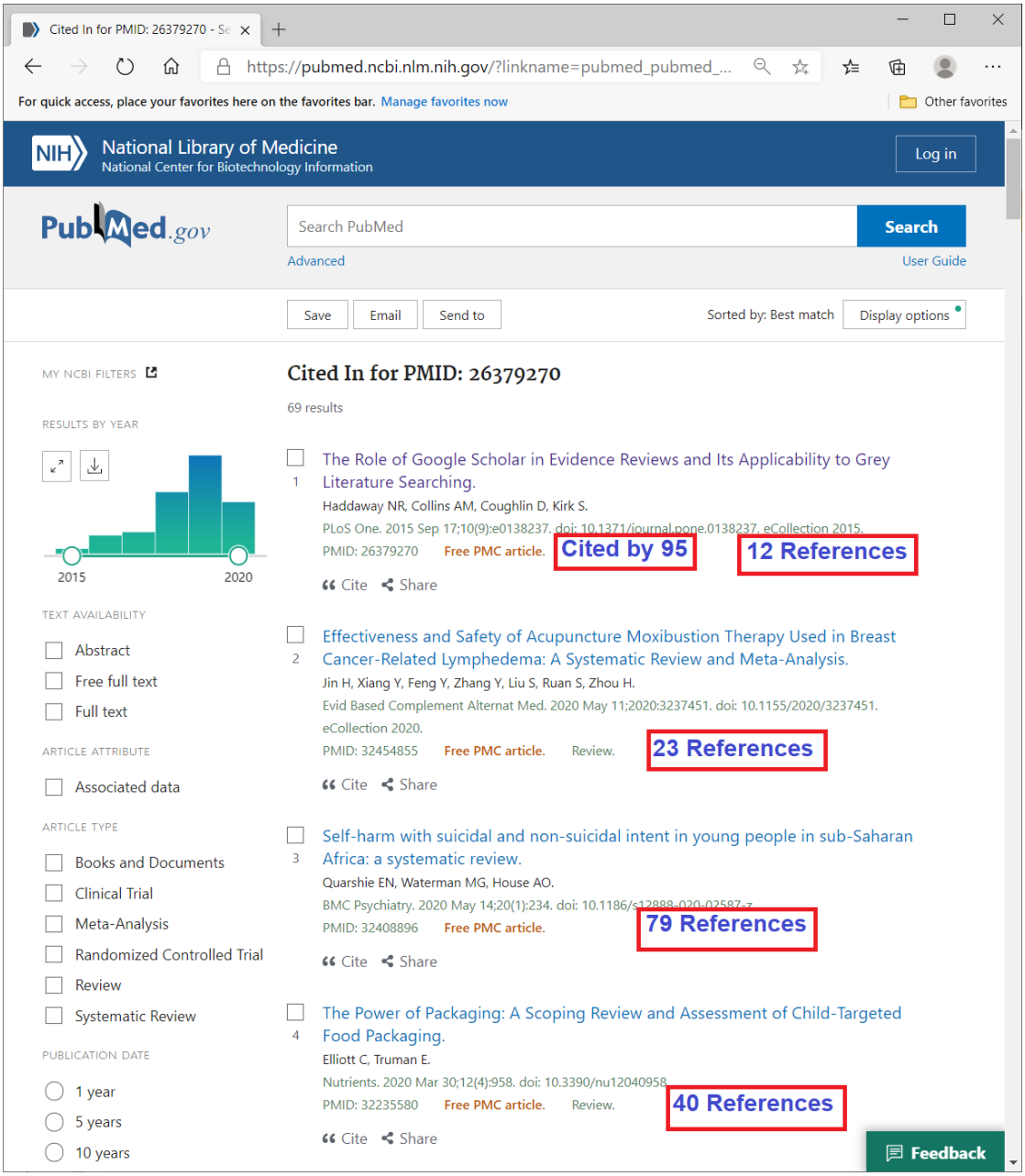

Figure 2.8: **The first three of 68 citations for a specific paper as I would like it to appear in PubMed.** I would like to see a *Cited by N* link and a *N References* link on the same line as the PubMed ID (PMID). Recently published citing papers (items 2 through 4) lack a *Cited by N* link because they do not yet have citations which are associated with a PMID.

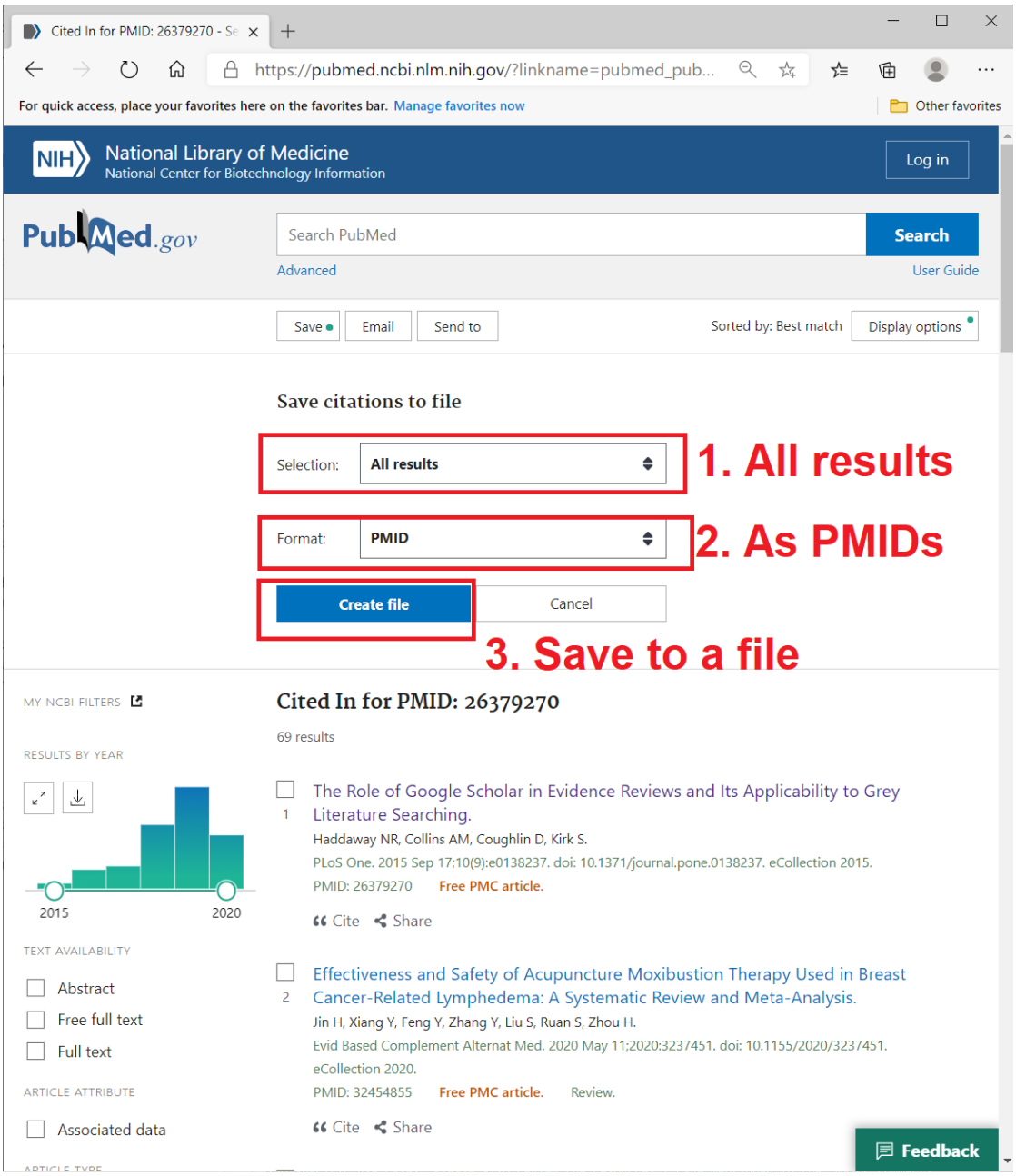

Figure 2.9: **Save PMIDs into a file.** Click the "Save" button. Then save all results as PMIDs into a file.

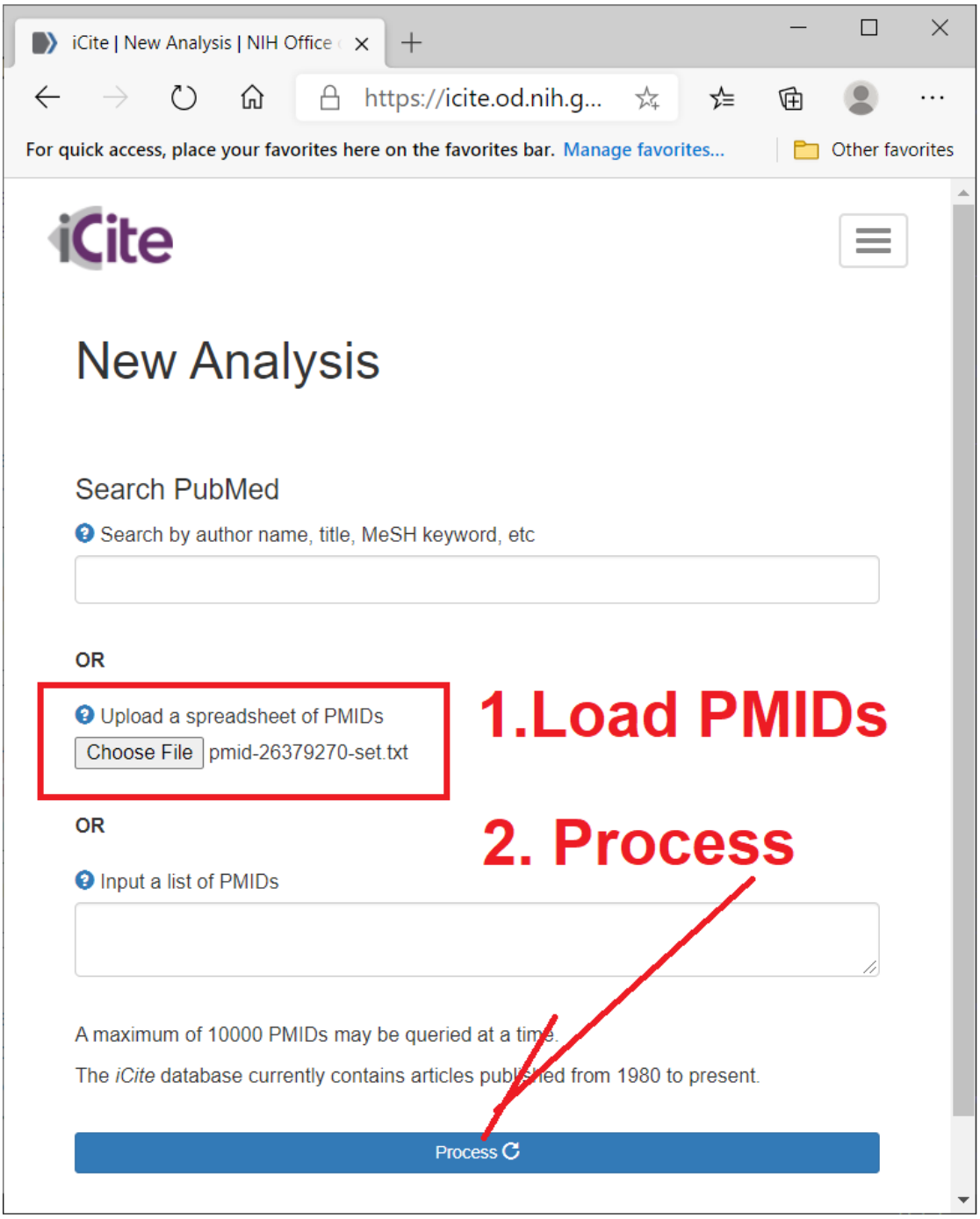

Figure 2.10: **Annotate PMIDs with NIH iCite citation data.** Load the PMIDs into iCite and process.

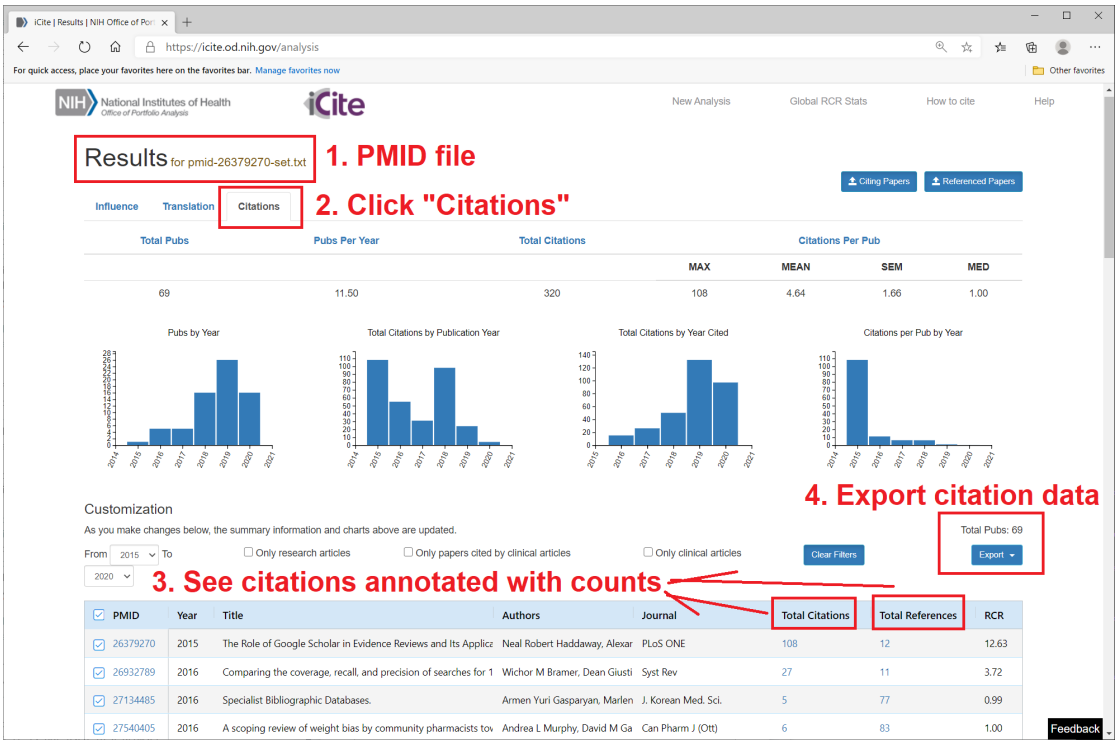

Figure 2.11: **Annotate PMIDs with NIH iCite citation count data.** Annotated citations appear as a list under the red "3" and do not retain their order sorted by PubMed's "Best Match" algorithm. The annotated citation data can be exported (red "4").

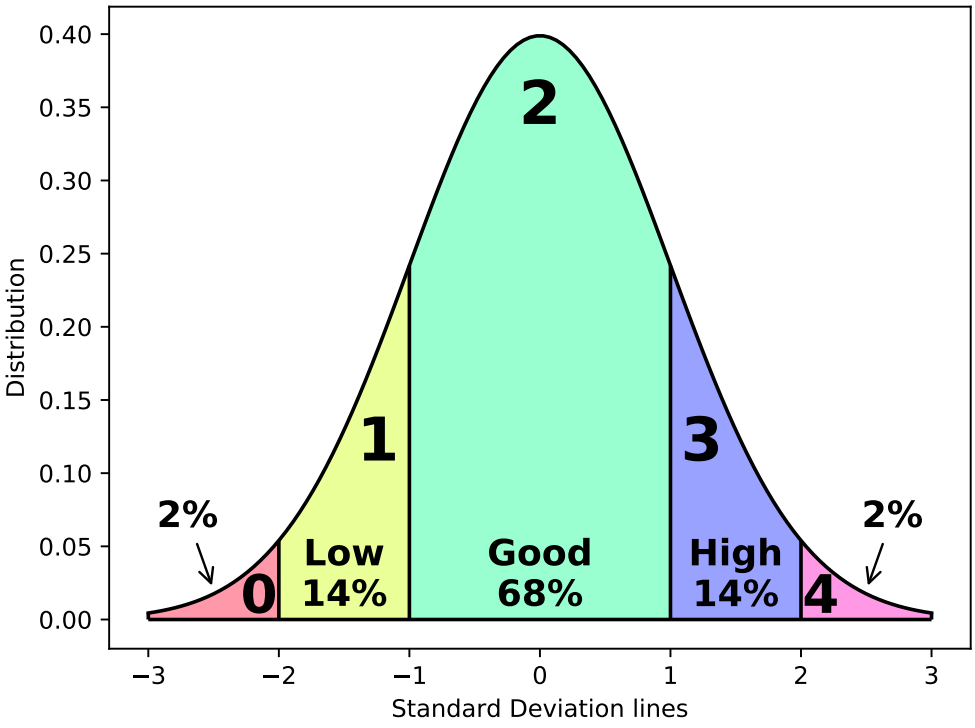

Figure 2.12: **NIH percentile groups.** There are 5 numbered NIH percentile groups. The lowest performing papers are given a 0. The highest performing papers are given a 4.

Chapter 3: DNA motif clustering algorithms

3.1 Gene clusters

Gene clusters are groups of genes located close to one another in the linear DNA sequence along a chromosome. They are sometimes the product of gene duplication events [139], and their expression is often coregulated [107]. The co-expression of neighboring genes has been found by genome-wide expression studies in humans [29], mice [172], and flies [195].

Coregulation may be caused by chromatin opening around one gene, allowing it be expressed, which may facilitate chromatin to be opened near neighboring genes [197]. Another reason for coregulation is that the genes in clusters may all be localized to the same nuclear compartment upon activation, as shown with the immunoglobulin and beta-globulin clusters. Another reason genes may cluster together is to facilitate staying together during meiosis, when the homologous chromosomes undergo genetic recombination [158].

Because gene clusters often share common gene expression, if a gene is found to be associated with a disease, the other genes in the cluster unassociated with disease may warrant further study. Uncovering genes not yet associated with disease can enhance understanding the molecular mechanisms of diseases and the identification of drug targets.

Overbeek defined gene clusters as a "run" or a group of collinear genes on a chromosome with no more than 300bp between them [159]. Overbeek's definition was used by the tool STRING (search tool for recurring instances of neighboring genes) to find sets of genes that are possibly functionally associated [193]. In this thesis, if Overbeek's 300 bp maximum intergenic

distance were to be used as the sole value to define a gene cluster, the middle part of the HOX cluster would indeed appear to be clustered (HOXA3, HOXA4, HOXA5, and HOXA6), but the upstream and downstream HOX genes (HOXA1, HOXA2 and HOXA7, HOXA9, HOXA10, HOX11, HOX13) are missing in that cluster. Using a maximum intergenic distance of 5k bp still leaves HOX9, HOX10, and HOX11 separate from the core HOX group found using a 300 bp distance. The result of using different maximum intergenic distance is explored in this thesis.

3.2 Two clustering algorithms

I describe two clustering algorithms used in identifying gene hotspots, which are based on defining the distance between DNA motifs. The first algorithm, called "start-to-start distance," defined the distance between two genes as the distance between the start of one gene to the start of its neighbor gene. The second algorithm, called "intergenic distance," defines the distance between two genes as the distance between two neighboring genes, which is the distance between the end of one gene and the start of its neighbor gene.

The first algorithm, "start-to-start distance," resulted in finding the densest clusters of motifs but at the cost of excluding long genes if the long gene was not near a short gene. In the human genome, gene lengths range from 147 bp to over 2Mb. The gene lengths for all investigated species skewed towards many more short genes than longer genes, with a median of 24kb for humans (Table 3.1). In the human genome, most genes (84%) were about 104kb long or less. For humans, the 16 keratin associated proteins (KRTAPP22-1, KRTAP19-2,

etc.) were the smallest genes ranging in size from 147 bp to 293 bp, while the longest human gene is "RNA binding protein, fox-1 homolog 1" (RBFOX1) with a length of almost 2.5Mb.

Table 3.1: **Gene length summary.** Gene lengths are skewed to having shorter rather than longer genes in humans, mice, and flies. The number of protein-coding genes listed in each genome is found under the quantity column, abbreviated as "qty." The standard deviation for all gene lengths is found in the right-most column "stddev." The 25th and 75th percentiles are found under the columns "25th" and "75th."

species	qty	min/max	25th	median	75th	mean	stddev
hsa	19,942	147-2.5Mb	8,447	24kb	65kb	64.5kb	130kb
mmu	21,907	156-3.2Mb	5,618	16.9kb	44kb	48kb	113kb
dme	13,911	168-2.0Mb	2,183	2.1kb	5kb	6.9kb	24kb

The second method "intergenic distance" resulted in finding the longest runs of genes that were tightly abutted to one another.

3.3 Methods

I use two methods to cluster genes: the "start-to-start" and the "intergenic" methods. Both are applied to the full set of protein-coding genes in a genome.

3.3.1 Genomic downloads

I downloaded genomic data from various sources including the University of Santa Cruz (UCSC), National Center for Biotechnology Information (NCBI) Gene [16], and Ensembl [91]. The background genomic topologies for humans, mice, and flies, contained in cytoband files were downloaded from UCSC[95].

The genome build and version for the cytoband data used in this thesis are shown in the table below. The species of human (Homo sapiens), mouse (Mus musculus), and fruit fly (Drosophila melanogaster) are abbreviated in this thesis as hsa, mmu, and dme respectively. The cytoband information contains a list of chromosome names for each organism as well as topological features, such as the location of each chromosome's centromeres and labels indicating small regions of each chromosome.

Table 3.2: **Species and genome builds.** Genome assemblies used for this thesis. An assembly is the full genome sequence determined by assembling with computer algorithms from small DNA sequences called contigs.

Species	Build	Date	Accession
hsa	GRCh38.p13	2019-02	NCBI:GCA-000001405.28
mmu	GRCm39	2020-06	NCBI:GCA-000001635.9
dme	BDGP6	2014-08	NCBI:GCA-000001215.4

Protein-coding genes and other DNA items for each organism were downloaded from two organizations, NCBI and Ensembl. NCBI is the United States organization established in 1988, whose collective biomedical databases compose the largest biomedical research facility in the world. Ensembl was launched in 1999 by the European Bioinformatics Institute and the Wellcome Trust Sanger Institute prior to the completion of the Human Genome Project. NCBI Gene focuses on fully sequenced genomes with an active research community whose contributions are updated in the NCBI Gene database as they become available.

Every day, the NCBI Gene database can experience incremental daily changes including the addition of new genes, the assignment of an official

gene symbol name, the addition of other names seen in the literature which are used for the gene, and an edit to the single official gene name. Ensembl gene also contains frequent updates to the official gene symbol name.

The full set of all genes in humans, mice, and flies were downloaded from the NCBI and Ensembl databases. Downloading the NCBI gene database is necessary because my queries for genes associated with disease used the NCBI Gene database. The Ensembl genome database is necessary for the three organisms since both NCBI Gene and Ensembl were data were needed for the ortholog studies.

The counts for protein-coding genes are about 20k genes for humans, 22k for mice, and 14k for fruit flies, as shown in Table 3.1.

3.3.2 Disease associations with genes

Six classes of major human disease were included in this research: cancer, infectious disease, autoimmune diseases, heart diseases, and environmental disease, as shown in the table below. There are 14 cancers in my disease list and 9 neurological disorders. The environmental disease class contains only one disease currently, which is obesity.

Disease Class	Key	Genes
Cancer	C	7133
Nervous System Disorder	N	3101
Autoimmune	A	2839
Heart	H	2290
infection	I	1871
Environmental	E	1667

Figure 3.1: **Six disease classes.** Individual diseases are grouped into six disease classes. Each class has a "Key", which is a single letter (C, N, A, H, I, E) used to identify the class succinctly. The number of genes for each disease class appears in the "Genes" column. There is one color per disease class: Cancer: purple; Nervous System Disorder: fushia; Autoimmune: Dark blue; Heart; Red; Environmental: Brown;

Disease	Key	hsa	mmu	dme	Disease	Key	hsa	mmu	dme
Lung Cancer	Ca	2459	226	1	Rheumatoid Arthritis	Aa	1097	141	1
Prostate Cancer	Cb	2413	287	0	Asthma	Ab	849	351	0
Hepatocellular Cancer	Cc	2248	261	1	Lupus	Ac	824	252	1
Colorectal Cancer	Cd	2230	187	1	Multiple Sclerosis	Ad	777	118	0
Gastric Cancer	Ce	1995	104	0	Crohns Disease	Ae	557	29	0
Squamous Cell Cancer	Cf	1897	115	4	Psoriasis	Af	535	117	1
Ovarian Cancer	Cg	1611	90	1	Ulcerative Colitis	Ag	528	56	0
Melanoma	Ch	1548	490	1	Pancreatitis	Ah	281	184	0
Glioma	Ci	1414	145	1	Diabetes Type I	Ai	3	0	0
Pancreatic Cancer	Cj	1270	126	0	Hypertension	Ha	921	434	1
Lymphoma	Ck	1144	312	9	Atherosclerosis	Hb	844	652	1
Renal Cell Carcinoma	Cl	891	39	0	Coronary Artery Dis.	Hc	732	12	0
Breast Invasive Cancer	Cm	6	0	0	Stroke	Hd	712	292	1
Schizophrenia	Na	1635	127	3	Heart Failure	He	590	315	2
Autism	Nb	549	103	10	Heart Disease	Hf	556	59	1
Alzheimer	Nc	519	148	3	Hepatitis C	Ia	1025	107	30
Parkinsons Disease	Nd	477	161	15	Hepatitis B	Ib	811	90	0
ALS	Ne	404	138	7	Tuberculosis	Ic	430	198	0
Mental Retardation	Nf	357	38	9	Influenza	Id	355	236	1
Major Depression	Ng	351	13	0	Malaria	Ie	206	130	1
Charcot Marie Tooth	Nh	97	25	2	Staphylococcus	If	90	67	3
Spinal Muscular Atroph	Ni	62	37	2	Obesity	Ea	1667	859	11

Figure 3.2: **Forty-two individual diseases This research included over 40 separate diseases, which are color coded by disease class.** The "Disease" column contains the name of an individual disease in this two-column table. The gene counts for human, mouse, and fly are identified under the columns, "hsa", "mmu", and "dme" respectively. The key for each disease is a two-letter combination. The first letter is the disease class, while the second letter represents the individual disease. There is one color per disease class and they are defined in the previous table.

Forty-two diseases and their associations with protein-coding genes were studied in this thesis, as shown in Table 3.2. To find the list of genes associated with each disease, one search per disease per species was performed using the

NCBI Gene website, located at `https://www.ncbi.nlm.nih.gov/gene`, for a total of 126 queries and downloads. I found genes related to numerous major diseases using "free text" queries for the diseases combined with specifying wanting only "live" or "current" gene entries for the species of Homo sapiens. The free text in the query could match text in various areas of the NCBI Gene entry, such as the title of the gene; the summary description of the gene; the titles of related PubMed articles; a GeneRIF (Gene Reference into Function), which describes the functions of the gene as verified by published experiments; entries in NCBI's Genetic Testing Registry (GTR); disease loci as identified in Genome-Wide Association studies (GWAS); etc.

3.3.3 Merge disease genes

Queries for over 40 diseases in humans resulted in over 35,000 mentions of a gene associated with a disease. Merging the gene lists from each query shows that this number is reduced to 9,254 unique genes. Investigating the number of diseases associated with each disease resulted in Figure 3.3 and the accompanying Table 3.3. This figure shows that most genes associated with disease are associated with only 1 to 4 of all diseases in my study. Genes associated with many diseases, such as over 17 diseases, comprise 295 or about 3% of the ~9k disease-associated genes.

The first set of genes to cluster is the set of all protein-coding genes. This merging of disease genes provides the second set of genes to cluster, which is the set of all disease genes. These two sets were clustered using two clustering methods to better understand the nature of gene clusters.

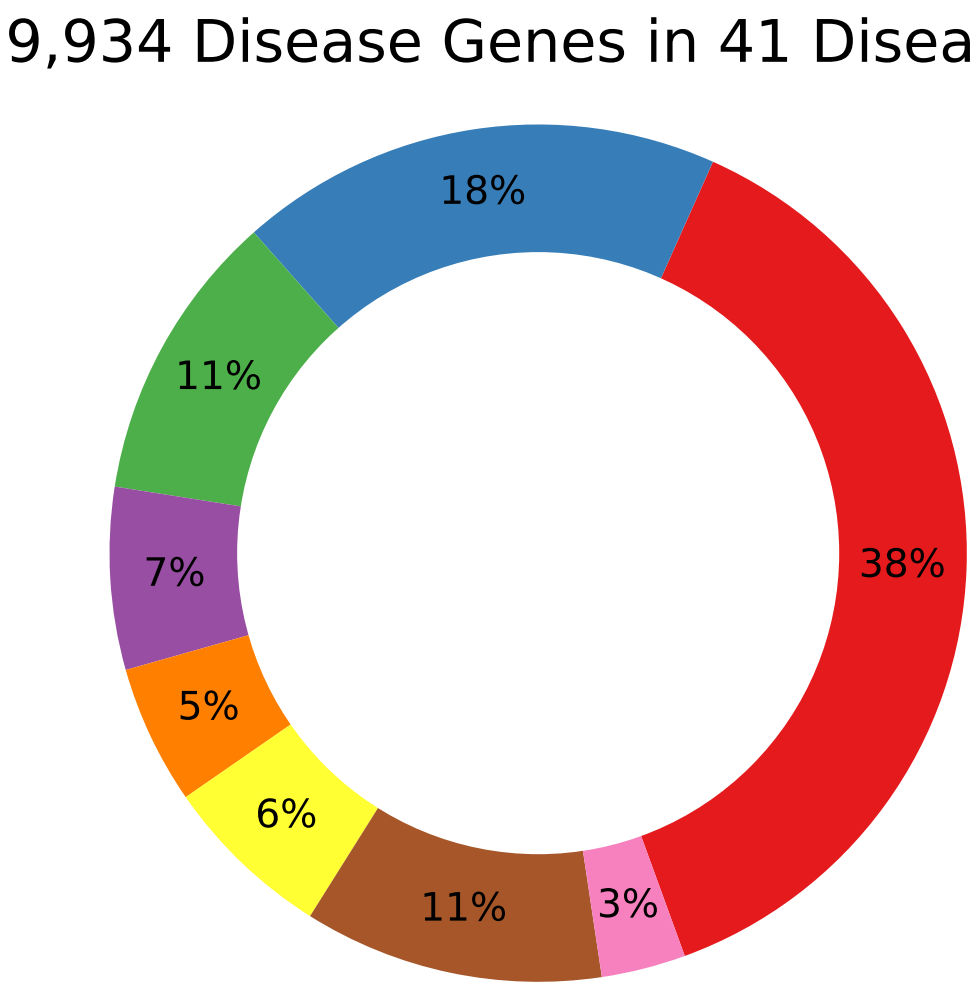

Figure 3.3: **Merged disease-associated genes for human.** There are over 9,000 disease-associated genes out of about 20,000 protein-coding genes.

3.4 Gene clusters: start–to-start method

Clusters containing the most densely packed genes on a chromosome are identified using agglomerative hierarchical clustering with the Euclidean distance metric and the average linkage. The Euclidean distance metric means that the distance between two genes is equal to the absolute value of the

Table 3.3: **Additional information about the figure above.** For example, 41% of the over 9,000 genes (or 3,748 genes as listed under '# Genes' column) are associated with only one disease, as listed under the '# Diseases' column.

Pie %	# Diseases	# Genes
41%	1	3748
18%	2	1701
11%	3	999
6%	4	589
5%	5	457
6%	6-7	557
10%	8-16	908
3%	17-41	295

starting point of one gene minus the starting point of another. Agglomerative hierarchical clustering is described below [144].

Agglomerative hierarchical clustering is accomplished with numerous iterations, where the first views each gene starting point as its own individual cluster containing only a single gene starting point. The last iteration results in a single cluster that contains all gene starting points. There are four steps for each iteration.

The first step determines the distance between the locations (starting points) of all genes and all other genes. In the "Iter 1" box in Figure 3.4 the distances calculated would be A-B, B-C, C-D, and D-E as well as A-C, A-D. and A-E as well all other distances. Distance calculations will be stored in a "distance matrix."

The second step is to form clusters for the two closest genes for the entire gene set. This is shown in each iteration box in the Figure 3.4 with the diagonal lines connecting the original clusters to make a combined cluster. The first pass

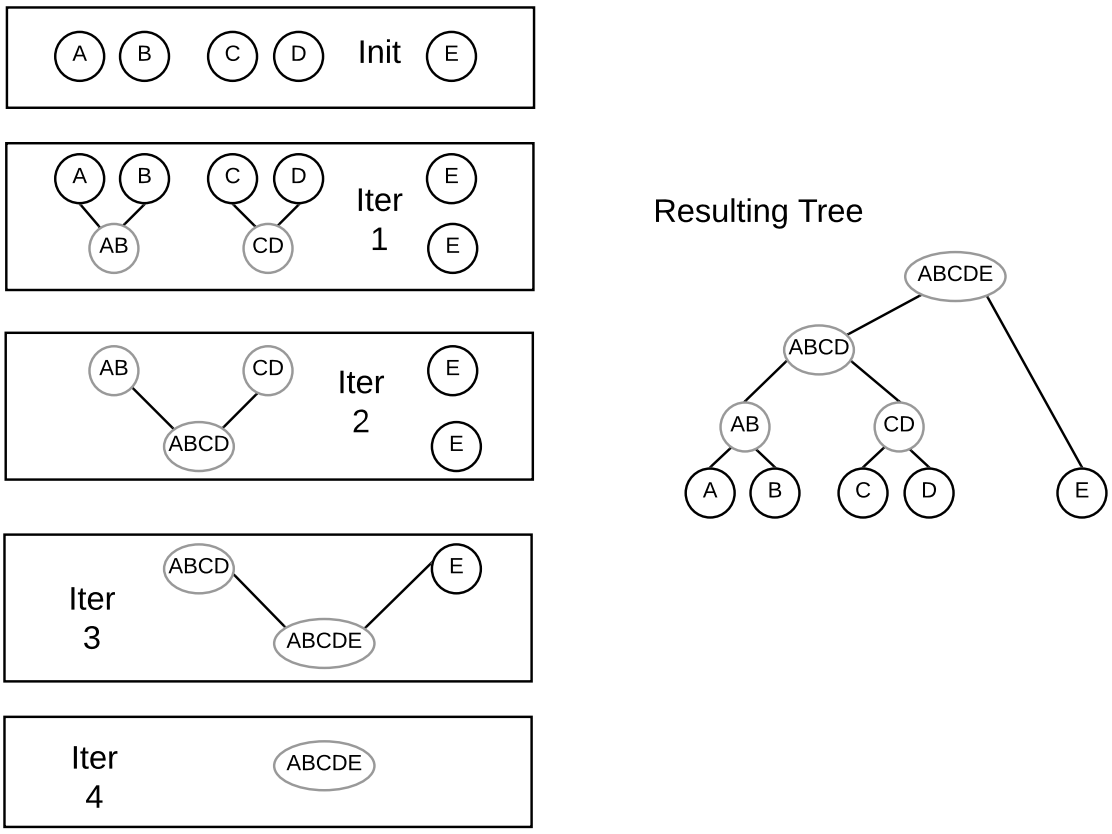

Figure 3.4: **Hierarchical Agglomerative Clustering.** Gene starting points are represented with A, B, C, D, and E. Intermediary nodes in the clustering process are represented as AB, CD, ABCD, and ABCDE.

of this algorithm will result in hundreds of 2-gene clusters on each chromosome for the most closely packed genes. The nodes AB and CD represent the new clusters of the closest gene starting points. A virtual "location value" will be calculated for each new cluster using the average linkage method. Average linkage means that if a cluster of two genes is created, the new cluster "location" will be equal to the average of the starting points of the two newly clustered genes.

The third step is to redefine the distances calculated from the first step so that the genes clustered during the second step are replaced by their new cluster, and the distance to all other genes is recalculated using the clusters' new virtual "location."

The fourth step is to return to the second step, except new clusters can be created not just from genes but from the new gene clusters created during the second step. These steps are repeated until only one top-level cluster remains.

Average linkage must be used for this project because the genes are collinear on a chromosome. Complete linkage is a tempting method since it is the default for the algorithm in Biopython [35], and the complete linkage provides a length value equal to the real cluster length, making this method initially seem preferable. For example, in Figure 3.5, both the average and complete linkages will create cluster "d," which contains 4 genes (A, B, C, and D). The length value calculated by the complete linkage is 8 (7-0+1), which is the actual length of this cluster. The length value calculated by the average linkage is 6, which is the location of the right-most intermediate cluster "i" (6) minus the location of the left-most intermediate cluster "h" (1) plus one.

The problem is that complete linkage will result in clusters that on their own, which appear to be desirable but are incorrectly splitting tight gene clusters at higher nodes, as shown in Figure 3.6. For example, genes G, H, I, K, and L should all form in a single cluster; however, when using complete linkage, they can be broken into two separate clusters "k" and "l," which are proper on their own, but dividing them is incorrect. Using average linkage provides the desired clustering, where genes G, H, I, K, and L are correctly placed in a single cluster "k."

This behavior is due to the nature of the algorithm and is not obvious or well known [49]. Collinear clusters becoming split when using complete linkage is not an initial condition problem since there are no initial random seeds used in hierarchical agglomerative clustering, and the result does not concern the data values.

I begin by creating a single hierarchical tree per chromosome. The final clusters of densely packed genes are obtained by making cuts in branches in the middle of the tree and by discarding the portion of the tree above the cuts. The cuts are chosen by determining those that will give the densest clusters whose total summed cluster length per chromosome is equal to a predetermined percentage of the length of the chromosome, such as 5%.

The cuts are determined using a "minimum density value," where the density of the cluster formed under each tree node in the cluster tree is calculated as the number of gene starting points divided by the length of the cluster. The nodes at the bottom of the tree tend to be the densest, while nodes at the top tend to be less dense. The middle nodes are usually less dense than their child nodes, but this is not always the case. The "minimum density value" is determined by placing the density of each node into a set, and the density list is sorted so that the densest value is first and the sparsest last. The next step is to iterate through this list, beginning with the densest value acting as the current "minimum density value," and through each node in the cluster tree, starting with all nodes in the leaf level and advancing up a level upon completing the row of nodes. A cut in the tree is made at the first node from the bottom to exceed the minimum density value.

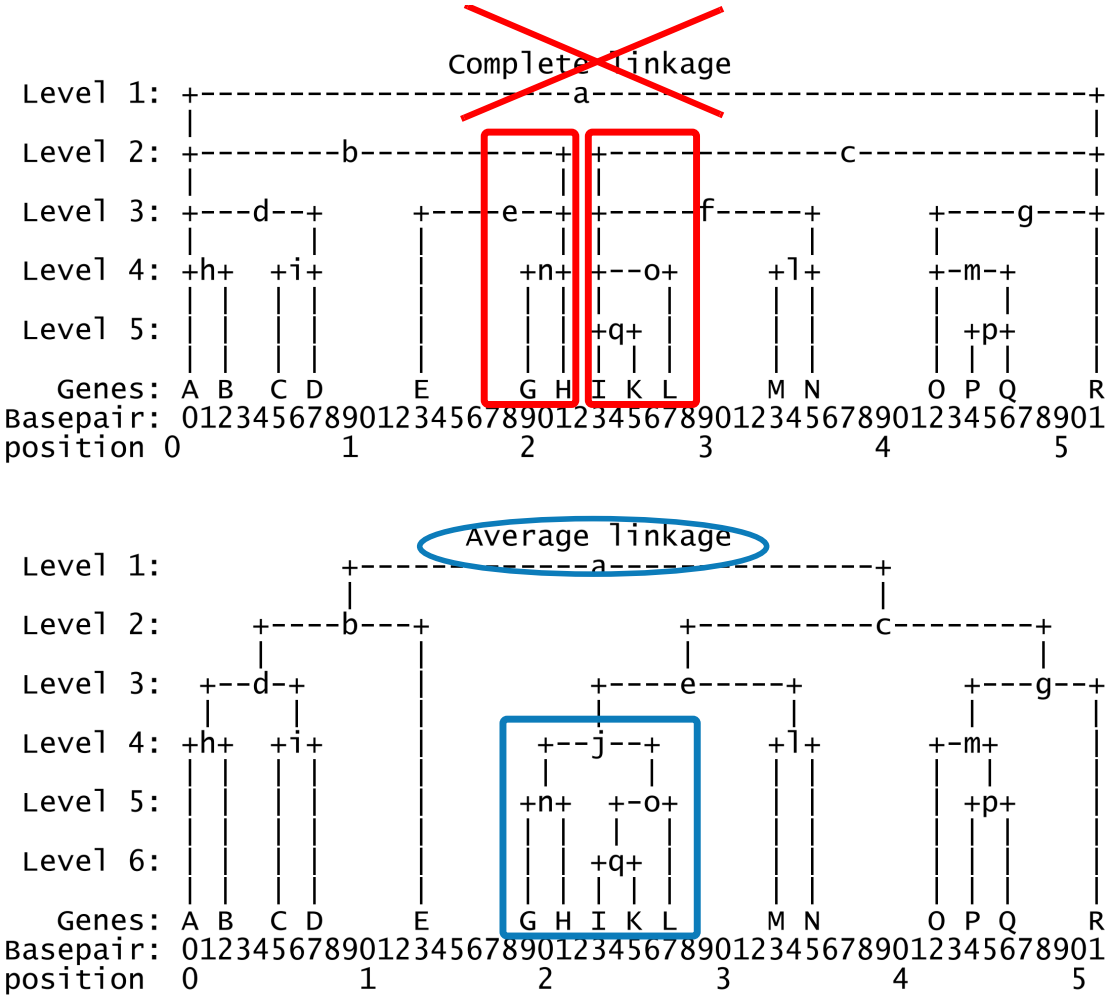

Figure 3.5: **Average linkage vs. Complete linkage for collinear data.** Average linkage is shown using the horizontal lines extending from the center of the left child cluster to the center of the right child cluster. Complete linkage is shown using the horizontal lines extending from the leftmost side of the left child cluster to the rightmost side of the right child cluster.

The next step is to create numerous sets of clusters, one set for each of the "percentage length" values ranging from 1% to 10%. To achieve the same "length percentage" for every chromosome, a different minimum cut density value must be calculated for each chromosome to determine a common total

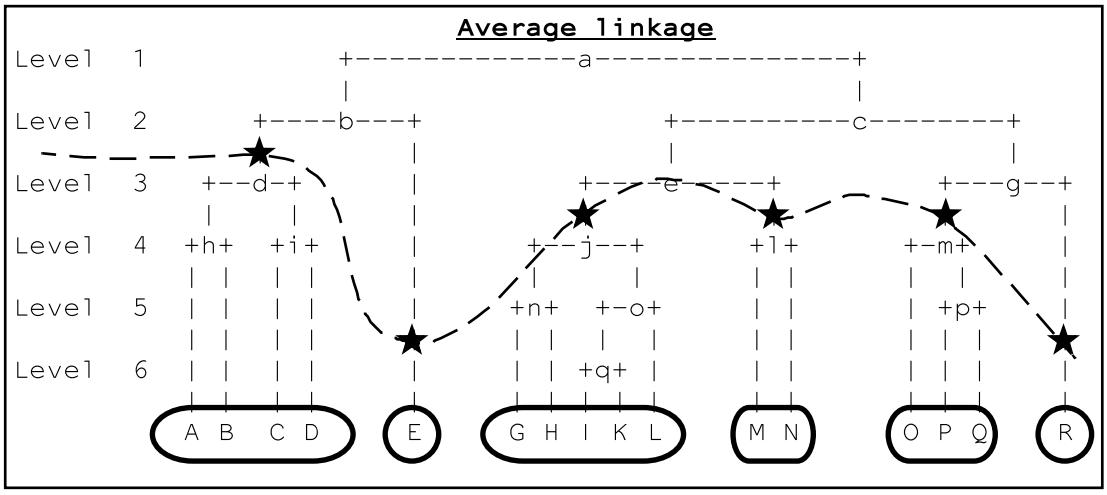

Figure 3.6: **Cuts in the tree result in gene clusters.** The stars represent cuts in the hierarchical tree. Cuts are made using a minimum density value so clusters created using the cuts are the densest layout of genes. Genes in the subtree below the cut are considered to be in the same cluster as shown by the ovals.

percentage cluster length for all chromosomes. The minimum cut density chosen, described below, is different for each chromosome since the protein-coding gene density for each of the twenty-three pairs of human chromosomes dramatically varies, from the densest chromosome Chr19 at 30.22 genes/Mb to the least dense chromosomes Chr Y at 2.13 genes/Mb and Chr13 at 3.48 genes/Mb. The chromosome lengths for the density calculations originate from UCSC's cytoBandIdeo.txt file from the most recent build of the human genome, GRCh38/hg38. The number of protein-coding genes for a single chromosome was downloaded from NCBI Gene in September, 2019. There were slightly less than 20k protein-coding genes with specific genomic coordinates, which is the chromosome name, starting base pair, and ending base pair.

Each chromosome must be separately considered to determine a minimum cut density, otherwise the calculated clusters would appear only on the most densely packed chromosomes such as chromosome 19, while any clusters on less dense chromosomes, such as chromosome 13, would consistently go unreported. Figure 3.7 below shows a bar graph of gene densities for each chromosome. The human genome is composed of 23 chromosomes: 22 autosomal chromosomes and the sex chromosomes X and Y.

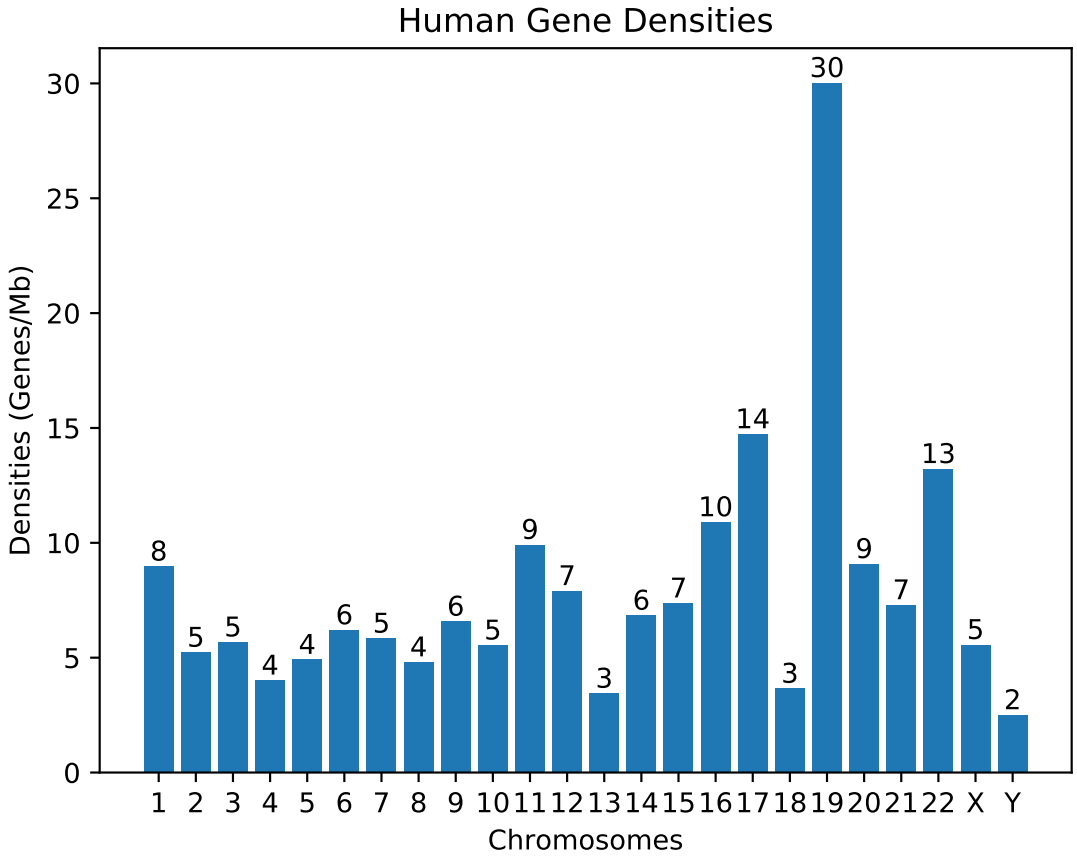

Figure 3.7: **Gene density bar graph.** Gene density per chromosome is calculated by dividing the number of protein-coding genes by the length of the chromosome according to the Genome Reference Consortium.

There are large areas of the genome which are "gene poor" and are shown circled in Figure 3.8. The human genome with protein-coding genes is drawn as small grey vertical lines in two rows per chromosome. The first row includes the protein-coding genes on the forward strand, while second includes genes on the reverse strand. The long grey-and-white rectangles extending from of each chromosome are the UCSC ideogram graphic, which aids biologists in describing an item's location on a long DNA strand. Gene-poor areas are commonly observed near many centromeres, drawn as crossed lines, and heterochromatic regions, colored in light cyan.

Although there can be a small number of genes in some of these gene-poor areas, because these areas were large in humans and generally had no or few genes, that length is subtracted from the total chromosome length to calculate the gene densities printed above.

The minimum densities values must be different for each chromosome, but the common parameter that ties together a set of minimum densities for all the chromosomes in the human genome is a percentage of the chromosome that is in a cluster and called the "percentage length."

The length of a cluster is determined by subtracting the smaller base pair value of the starting point of the left-most gene in the cluster from the larger base pair value of the starting point of the right-most gene, regardless of the strand where the genes are located. The density of a cluster is calculated by dividing the total number of genes in a cluster by the cluster's length. The tree is cut using a minimum density and keeps the nodes just below the cut point. The nodes just below this point will be the cluster set for the minimum density

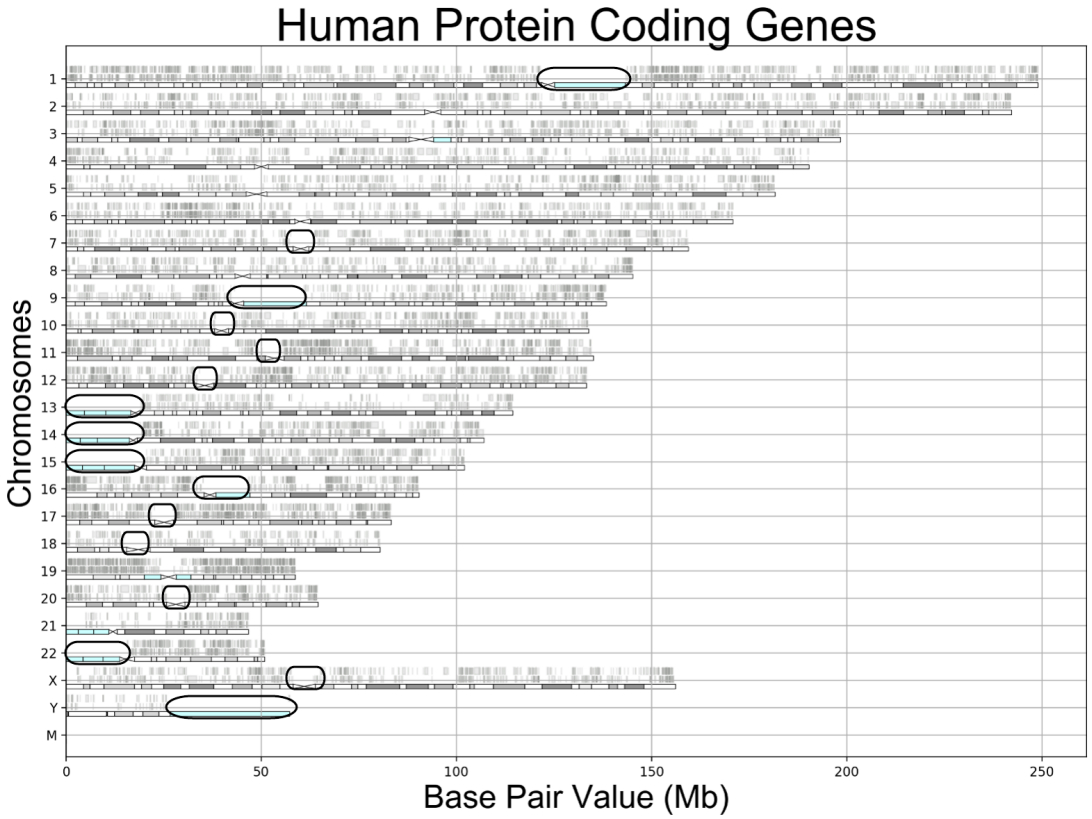

Figure 3.8: **Human protein-coding genes with gene-poor areas circled.** Gene-poor areas are frequently found near centromeres, acrocentric bands near centromeres, and heterochromatic areas of the chromosomes.

set. The percentage length for this set of cut-points is the sum of the lengths of the nodes at the cut point divided by the length of the chromosome.

If the tree is cut for a 1% summed cluster length for the case of all disease genes, there will be about 3,000 tiny clusters. Many these clusters are dense two-gene clusters. A few clusters have more than ten genes and are of interest for this study, and some two-gene clusters may be of interest if sitting closely together. If a two-gene cluster is alone with no close neighbors, it is not of interest for this study.

The clusters containing ten or more genes can be expanded by aggregating some of the two-gene clusters together by considering the 5% summed cluster length, which will contain all genes in the 1% summed cluster length, but some of the small individual 1% clusters will be combined into a single, larger cluster. This results in expanding the 1% clusters containing 10 or more genes to be longer and encompass more genes as well as aggregating many two-gene clusters into a larger cluster. The undesirable effect of using a larger summed length such as 5% is that many new clusters are created which are less dense and not of interest in this study.

3.5 Gene clusters: intergenic method

The start-point clustering method will find areas in the genome with the most genes per Mb. This method favors small genes at the cost of excluding long genes if they do not surround the small genes. The intergenic method seeks the longest runs of genes with little intergenic space between neighboring genes.

The intergenic method is implemented as follows: The first step is to choose a maximum distance. Genes will not be considered in the same cluster if their intergenic distance from one another exceeds the maximum intergenic distance. The genomic coordinates between a gene's starting and ending points is considered to be its interval. For each chromosome, the genes' intervals on both the forward and reverse strands are joined using the union operation. By performing a union operation on all genes, any gene overlaps are collapsed so that all resulting intervals do not overlap with others. The new intervals formed by the gene unions are then walked starting at the

interval closest to base pair zero, shown on the left side of genome figures, to the last interval on the chromosome, shown on the right side of genome figures. A new "current cluster" is created with starting and stopping points set to those of the current interval. If the current interval is not within the maximum distance of the next interval, the current cluster is saved as one finished cluster; if the current interval is within the maximum distance of the next interval, then the current cluster's ending point is changed to the next interval's ending point. The next interval becomes the current interval and the walk is continued along the upcoming intervals, and this process is repeated until all intervals are placed into clusters.

The next step is to identify the list of genes used to create the cluster. Clusters which contain only one gene are removed from the cluster list and placed into a "lone gene" list.

The larger clusters containing 10 or more genes are retained for further study. Numerous sets of clusters are created using various maximum distances.

3.6 Results

Using the start-to-start method to create clusters so that the summed length of all the densest clusters is equal to 5% of the total genome length, these clusters will contain between 55%-65% of the genes used to create the clusters, as shown in Table 3.4.

Table 3.4: **Total cluster length and percentage of genes in clusters as clustering parameters are varied.** The percentage of genes found in clusters rises rapidly when evaluating clusters created using total cluster lengths of 1% and 5% of the genome, but it then slows after 5%.

Species	Qty	Genes	1%	2%	3%	4%	5%	10%
hsa	19,942	protein-coding	36%	45%	51%	56%	59%	72%
hsa	9,374	disease genes	30%	40%	46%	51%	55%	70%
mmu	21,907	protein-coding	33%	43%	49%	54%	58%	69%
dme	13,911	protein-coding	40%	50%	56%	61%	65%	77%

The densest clusters which comprise a total 1% and 5% of the total genome length appear in Figure 3.10, which shows about 3,000 and 3,700 clusters for 1% and 5% of the genome length respectively. The clusters are sorted by length so that the smallest clusters are on the left and the largest on the right. The cluster density is shown above its cluster using a purple dot, and the number of genes in each cluster is shown above its cluster as a blue dot. The bottom line of blue dots for the 1% clusters shows that most of the clusters at 1% genome length are 2-gene clusters ranging in density from above 500 genes/Mb to under 20 genes/Mb. For clusters comprising 5% of the genome length, a small percentage have ten or more genes in a single cluster. The densest clusters have ten or more genes since they stand out among thousands of clusters and are circled by the magenta box.

Figure 3.9 shows the densest protein-coding gene areas that form clusters covering 5% of the genome, and almost all these clusters have less than 10

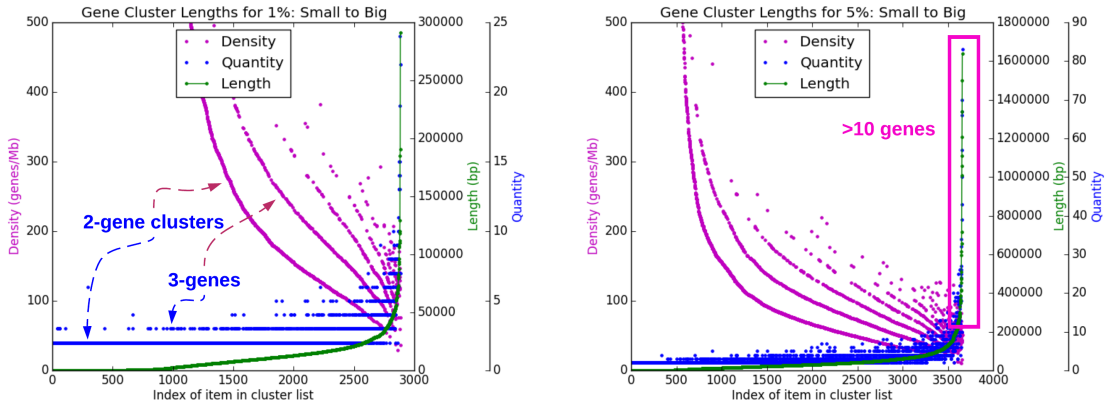

Figure 3.9: **Cluster characteristics at 1% and 5% percentage length.** This figure shows the relationships between cluster length, density, and gene count on a chromosome. Thousands of clusters are sorted by length with the smallest clusters on the left and largest on the right. At 1% total genome length, the smallest clusters tend to have two or three genes (blue dots) and are exceptionally dense (purple dots). Few clusters contain 10 genes or more (rightmost side). At 5% total genome length, the smallest clusters have less than ten genes (blue dots).

genes. The most densely packed clusters with 10 or more genes are studied in this thesis.

The clusters to be retained have 10 genes or more and contain at least two genes within that are exceptionally densely packed, as shown by the 1% genome length cutoff. After removing small clusters and those whose genes are spaced sufficiently wide so that none of their genes are contained in the 1% genome length clusters, the remaining clusters are shown in Figure 7. The figure also shows areas which are dense for protein-coding genes, which are associated with disease.

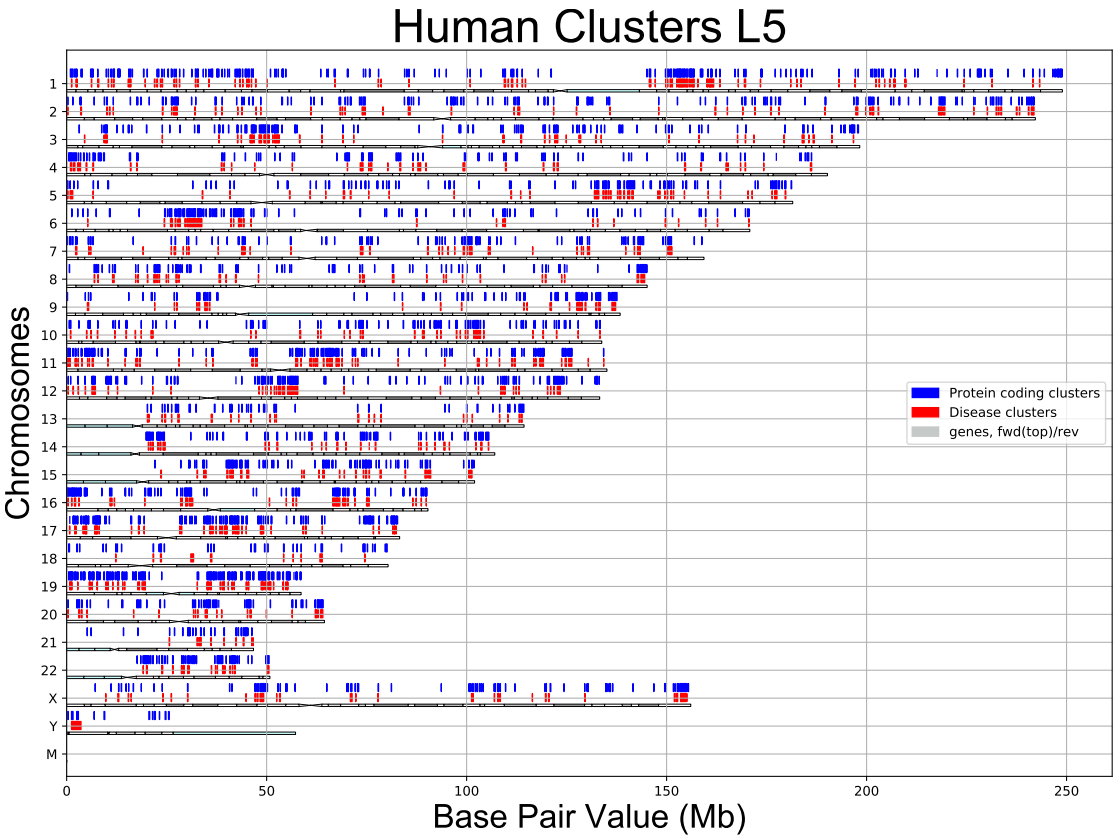

Figure 3.10: **All clusters comprising 5% of the genome length.** Over 100 of these clusters have 10 or more genes, while the majority have only two or three genes.

3.7 Discussion

The human chromosomes drastically vary regarding their content, especially gene-poor areas which include centromeres, “gvar,” and “stalk” regions.

In UCSC’s cytoband data, gvar regions are especially tightly packed DNA regions known as variable heterochromatic regions, which are either pericentric (surrounding the centromere) or telomeric (located at the ends of chromosomes).

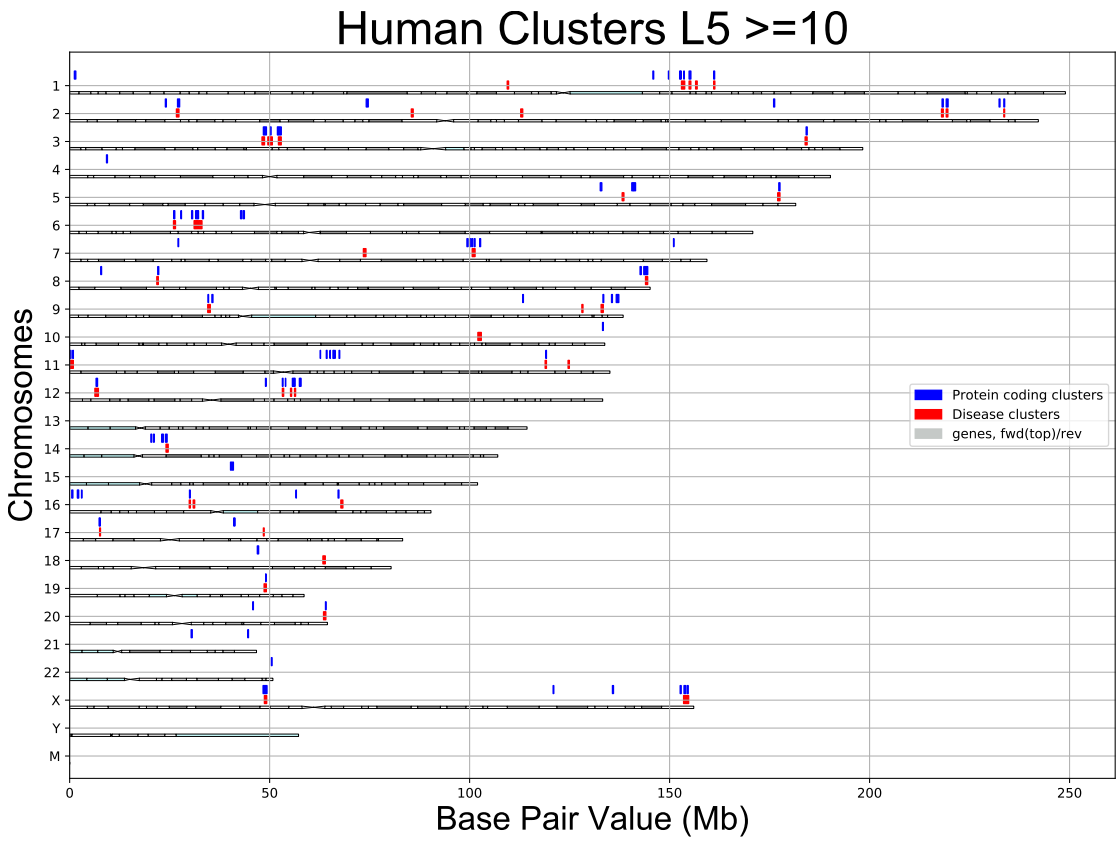

Figure 3.11: **Largest, densest clusters comprising 5% of the genome length.** Clusters for all protein-coding genes are shown in blue. The subset of protein-coding genes associated with disease are shown in red. The genomic areas which contain the most densely packed disease genes are not always in the same area as generally dense areas of protein-coding genes.

Stalk chromosomal regions, appearing on only chromosomes 13, 14, 15, 21, and 22, are thus named due to their skinny, stalk-like appearance when chromosomes in dividing cells are viewed through a microscope.

Due to the large size of the gene-poor areas and large differences in their quantity among different chromosomes, quantifying gene density involves subtracting these areas from the length of the chromosome.

The top five genes associated with most diseases are interleukin 6 (IL6), tumor necrosis factor (TNF), vascular endothelial growth factor A (VEGFA), transforming growth factor beta 1 (TGFB1), and nuclear factor kappa B subunit 1 (NFKB1). The genes IL6, TNF, NFKB1, TGFB1 are especially associated with functions of immunity.

Applying the cluster methods start-to-start and intergenic to the same sets of genes often results in finding overlapping clusters, but not always. The start-to-start agglomerative clustering method results in numerous clusters where disease and protein-coding clusters do not overlap. The intergenic clustering method always has disease gene clusters as a subset of protein-coding clusters since I use the same maximum distance for each. As expected, the densest chromosome chr19 quickly becomes covered with clusters even at a maximum distance of 10kb, while all other chromosomes exhibit sparse and separated clusters.

3.8 Conclusion

Although one might expect many of the same genes to be found in many cancers, if a gene merge is performed on only the ˜3k genes associated with the 14 cancers, there is a similar pattern as the gene merge performed on all disease genes, because most cancer genes are only associated with one cancer. About 100 cancer genes are associated with 8 or more cancers. The pattern is the same regarding neurological diseases from over 40 diseases under consideration. It can be concluded that a small percentage of genes are

associated with large numbers of disease, whether general disease across all categories or within specific categories such as cancer or neurological disease.

It was expected that the clusters from the two cluster methods would have similar results in many cases and different results in the long gene clusters. Applying the two methods provided a richer understanding of how disease genes cluster in the genome.

Chapter 4: Protein-coding hotspots in the human genome

4.1 Introduction

The goal of this thesis is to identify gene-rich regions of DNA and which genes in the clusters have similar functions as other genes in the same cluster but are not in the same gene family.

Previous studies indicate the presence of spatial clustering of genes, assuming similar and/or complementary functions in bacteria and Archaea [159]. It has long been known that genes cluster together in bacteria; in 1959, Demerec and Hartman suggested that natural selection must keep the gene clusters together, and thus the gene clusters must be beneficial both to the individual and their population [43].

Gene clusters are kept together in a single species over time and stay together even when new species are formed in a speciation event from a common ancestor. My studies of orthologous protein-coding genes show agreement that gene clusters in humans near several nuclear lamina domains also cluster together in mice. Nuclear lamina domains are areas of the chromosome which frequently contact the nuclear membrane encasing the nucleus [104]. Genes found on the periphery of the nucleus are generally less transcriptionally active than genes found inside the core of the nucleus [174]. A notable feature of gene clusters is that they are often composed of functionally related genes [159], and Overbeek proposes that conserved gene clusters across different organisms can be used to predict functional clustering.

Gene families, originating from a common ancestor, typically form spatial clusters on eukaryotic and prokaryotic genomes. An example is the G

Protein-Coupled Receptors (GPCR) that originate from a common ancestor in the human genome and are grouped in the main families of rhodopsin, secretin, glutamate, and frizzled [64].

Small regions of genomes can be identified as being enriched with DNA motifs involved in developing of multiple complex diseases. An example is the Human Leukocyte Antigen region (HLA), also known as the Human Major Histocompatibility (MHC) genomic region, which is located in the chromosome genomic location 6p21. The HLA complex aids the immune system in recognizing which molecules are foreign and which are the self [53] so that pathogenic foreign substances spur an immune reaction, while internally created proteins do not invoke an immune reaction. The HLA region is involved in many diseases ranging from autoimmune disorders to heart disease and cancer [190] [109]. My analysis aligns with previous findings of high disease association in the HLA region at 6p21, finding that genes in that region are associated with over 40 diseases studied in this thesis.

The recent accumulation of annotated genomes in the open access domain enabled researchers to create inquiries about the statistical organizations of genomes and their comparisons across genomes. Annotated genomes refer to the mapping of various DNA elements onto a specific chromosome and base pair locations. Elements that have been mapped to a specific location on genomes include long noncoding RNAs by the GENCODE project [82], which have been found to regulate gene expression in cis-acting (nearby on the same chromosome) and trans-acting (across chromosome) modalities. Some examples of regulatory control of long noncoding RNAs include regulating metabolic control [219] and glucose homeostasis [181].

Like the microRNAs discovered in the early 1990s, noncoding RNAs regulate gene expression after a gene has been transcribed from DNA, which has been known since the early 2000s. In 2012, the ENCODE project released the loci of 9,640 long noncoding RNAs [82]. Long noncoding RNAs (lncRNA) control gene expression before and after (like microRNAs) transcription in numerous ways [205]. lncRNAs are also unique since they can directly interact with DNA as well as proteins and RNA.

Long non-coding RNAs have recently been shown to be important in regulating glucose metabolism [181], which is highly regulated in mammals and involves numerous molecules including transcription factors, enzymes, and transporters. Misregulation in the metabolism of glucose has been linked to diabetes and cancer [26].

Enhancers are also important cis-acting regulatory elements [168], are currently thought to number in the hundreds of thousands, and can be located upstream, downstream, or in introns of a gene that they control. Enhancers that are mapped onto the genome can be examined regarding proximity to genes associated with disease. For example, Dickel et al. recently mapped over 80,000 putative human heart enhancers onto the genome [47], and relational databases for genes also emerged, which linked the gene symbol to its chromosome position, sequence, function, protein-binding partners, and related literature. Unifying data from drug databases, gene databases, and disease databases can provide insights into biological issues [102].

4.2 Disease in the human genome

Uncovering protein-coding genes not yet associated with disease can aid understanding the molecular mechanisms of diseases and potentially reveal novel drug targets. Previous studies of disease and genetics have included population genetic linkage analysis to determine common variants of genes involved in a specific disease condition and to compare allele frequencies among major ethnic groups. Another approach analyzed multiple diseases using GWAS results [69]; however, these studies did not focus on the topology of many major diseases on the human genome. I studied the topology of the human genome to find the densest clusters of genes (i.e., hotspots) associated with disease using two sets of hotspots. Clustering all protein-coding genes associated with disease creates the first set of hotspots, while clustering all protein-coding genes creates the second. I then seek shared genomic hotspots to highlight genes not yet associated with disease for further study.

4.3 Disease across animal models

Identifying and understanding the differences and similarities in protein-coding genes and their gene expression in various species can provide new information regarding gene functions and their role in various major diseases, which aids in identifying gene targets for possible drug testing. Choosing model organisms essential to general medical research, such as the Mus musculus (house mouse) and Drosophila melanogaster (fruit fly), can expand the ability to study shared genes and their expression. Recent studies have found that humans and mice share a majority of protein-coding genes

[218], and that gene expression in the organs of humans showed similarities to gene expression in the same organs of other vertebrates [200].

Compared to organisms, such as the mouse, the fruit fly is a time- and cost-effective model organism for researching biological processes involved in genetic disease [211]. The fruit fly has almost 14k protein-coding genes compared to humans with almost 20k. An analysis in 2001 of about 900 human genes linked to disease estimated that 75% of human disease genes have homologous genes in Drosophila [175]. As of September 2019, about 9.25k human genes are linked to over 40 major diseases out of 20k human protein-coding genes. Research using the fruit fly has revealed factors used in chromatin maintenance, which is critical in epigenetics that control gene expression in fruit flies and humans [178]. Research with the fruit fly has revealed important signaling pathways controlling the development of the organism and determination of a cell's fate [30]. Fruit fly research has been important in neurological disorders since there are important similarities in gene expression between the brains of fruit flies and humans. The fruit fly has already been used to study Alzheimer's, Parkinson's, and Huntington's diseases [150] and amyotrophic lateral sclerosis (ALS) [194].

4.4 Hotspots for protein-coding genes and disease-linked genes

This section contains the chromosome maps of the human genome hotspots for protein-coding genes and for their subset of disease-linked genes, and it highlights the understudied genes as determined by the IDG group. The number and size of hotspots change by varying the maximum allowable

distance between adjacent DNA motifs in a hotspot. The gene membership of each hotspot is annotated in tables along with their linkage to human disease obtained by inquiries using the NCBI Gene database. Disease-linked gene hotspots are not uniformly distributed on the genome but have highly dense hotspots in some chromosomes such as 1, 3, 6, 11, and 12.

4.5 Protein-coding clusters and disease-gene clusters

Two clustering methods, hierarchical agglomerative clustering and intergenic clustering, performed on the full set of ~20k protein-coding clusters and ~9.5k disease-gene clusters, resulted in four sets of clusters, which are summarized in Table 4.1. The ~20k human protein-coding genes cover ~21% of the human genome, including both introns and exons. For clusters studied in this thesis, between ~5% and ~9% of all protein-coding genes lie in the studied clusters. The chosen sets of clusters cover between 0.79% and 1.14% of the human genome.

Table 4.1: **Selected clusters.** The "Key" column holds an alias representing the cluster set. The "Method" column indicates the clustering method, which is either agglomerative clustering (AG) or intergenic clustering (NN). The "Gene set" column indicates the set of clustered genes: All ~20k human protein-coding genes or the subset of ~9.5k protein-coding genes associated with disease. The "# clusters" column indicates the total number of clusters containing over ten protein-coding genes or disease genes. The "# genes" column indicates the total number of genes contained within all clusters for the cluster set. The "Cluster length" indicates the total genomic length taken by all clusters in the cluster set. And the "Genes" column indicates the total percentage of protein-coding genes which are contained in all clusters in the cluster set.

Key	Method	Gene set	# clusters	# genes	Cluster length	Genes
P	AG	Protein-coding genes	107	1747	0.86%	8.76%
D	AG	Disease genes	48	1055	0.79%	5.29%
p	NN	Protein-coding genes	117	1569	1.15%	7.87%
d	NN	Disease genes	56	1023	0.91%	5.13%

4.6 Choosing cluster parameters

Using agglomerative and intergenic clustering methods to explore the genome enables comparing results. The agglomerative clustering used starting points of genes resulting in the clustering of genes whose starting points were extremely close together. The concern was that agglomerative techniques favored clustering short genes and ignored long genes unless they were found on the edges of the cluster. In contrast, intergenic clustering does not inherently exclude long genes.

The parameter that can be varied in the agglomerative clustering, which affects the number and size of the clusters, is percentage length. Upon walking the percentage length parameter from 1% to 20%, using 5% provided the best ratio of the shortest clusters containing the largest number of genes. Using a 5% percentage length resulted in thousands of clusters, with most being two-gene clusters. The clusters for this thesis were limited to those containing ten genes, which resulted in the 107 protein-coding clusters and 48 disease clusters.

The parameter that can be varied in intergenic clustering is the maximum intergenic space between two neighboring genes. Upon walking the maximum intergenic distance from 2k base pairs to 200k base pairs, the most comparable results to the agglomerative clustering method regarding clusters found to the agglomerative clustering method is a maximum intergenic distance of 10kbp for all ~20k protein-coding genes and 50kbp for clustering disease-associated genes. Using 50kbp for clustering disease genes resulted in clusters containing genes closer than 50kbp, because some genes not used for clustering but still contained within the cluster had no association with disease.

4.7 Comparing the four cluster sets

Four cluster sets are shown in Figure 4.1, each indicated by a different color. The two agglomerative cluster sets are indicated by the dark blue bar for protein-coding clusters and red bar for disease clusters. The two intergenic cluster sets are the light blue and magenta bars, which indicate protein-coding clusters and disease clusters respectively.

There is concurrence in the clusters appearing in the large groupings of clusters in the areas surrounding 1q21.3, 3p21.31, the HLA clusters in the area of 6p21.33, and 11q13.1, 12q13.2, with about 44% of the genes observed in both the agglomerative clustering and intergenic clustering.

A notable difference is that disease clusters do not always occur in protein-coding clusters. The chromosome showing the most difference in clustering results between the two clustering methods is chromosome 19, the densest chromosome. The densest clusters on each chromosome are found by normalizing for each chromosome when using agglomerative clustering. For intergenic clustering, no normalization per chromosome was used, resulting in large numbers of clusters covering much of chromosome 19.

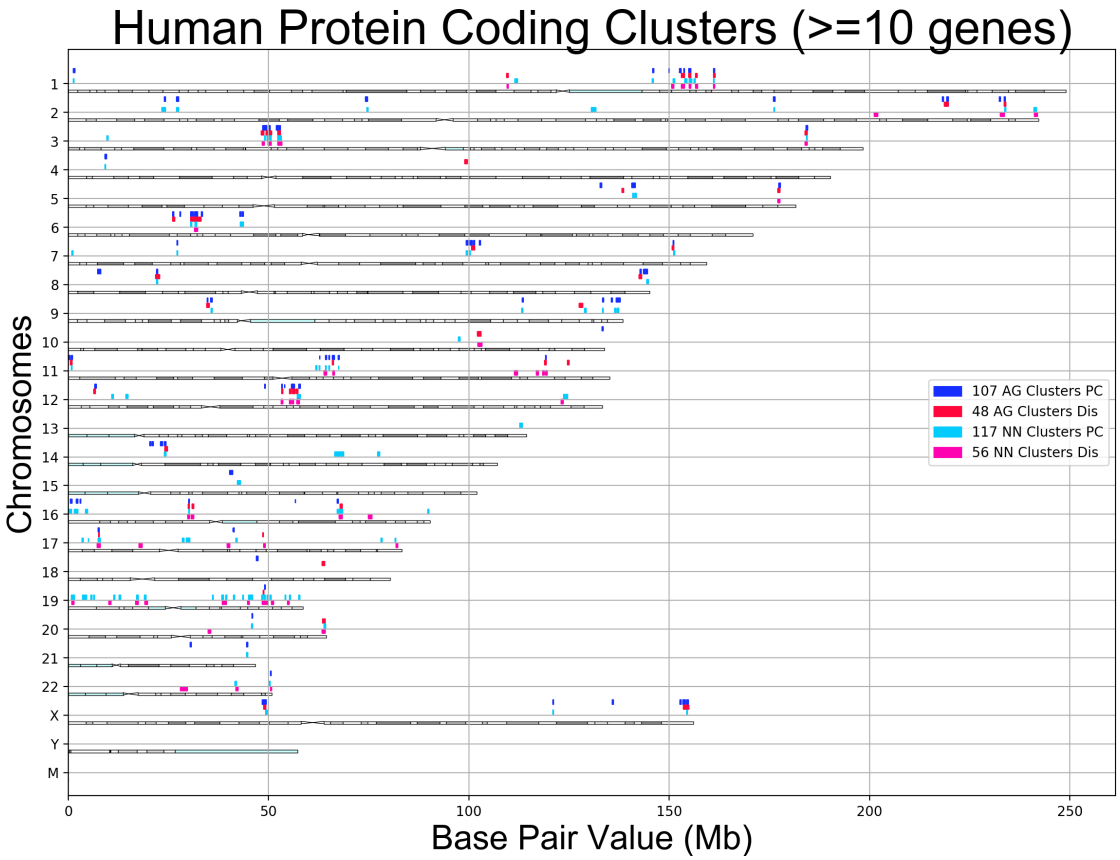

Figure 4.1: **Four sets of clusters of protein-coding genes on the human genome.** Clusters containing ten or more genes on the human genome. Four sets of clusters are shown: 107 clusters of protein-coding genes (PC) created with agglomerative clustering; 48 clusters of disease-associated genes (Dis) created with agglomerative clustering (AG); 117 clusters of protein-coding genes created with intergenic clustering, also known as nearest-neighbor clustering (NN). There are 56 clusters of disease-associated clusters created using intergenic clustering.

The total cluster lengths are similar between the two clustering methods in Figure 4.2. The protein-coding cluster distribution is similar for both agglomerative clustering (labeled Pro-AG) and intergenic clustering (labeled Pro-NN). The clusters created using disease genes show similar cluster length

distributions between the agglomerative clustering (labeled Dis-AG) and intergenic clustering (labeled Dis-NN). The clusters created using disease genes are larger for both methods since the clustering parameters are loosened when creating the clusters due to the smaller set of genes in the total disease population.

The median size of the protein-coding clusters is ~209 kbp for agglomerative clustering and 244 kbp for intergenic clustering. The median size of the disease clusters is 437 kbp for agglomerative clustering and 461 kpb for intergenic clustering.

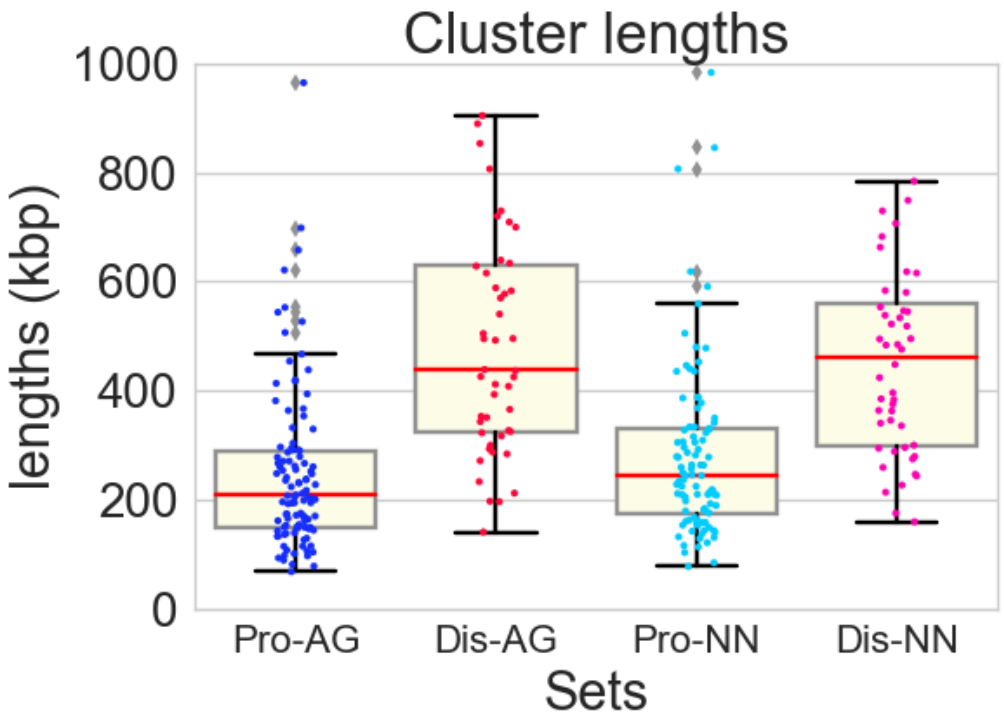

Figure 4.2: **Cluster-length distribution of the four cluster sets.** The distribution of cluster lengths is shown. The protein-coding cluster sets are named on the x-axis with the name Pro. The disease cluster sets are named using Dis. Clusters created with agglomerative clustering are named on the x-axis using AG. Intergenic, also known as nearest neighbor cluster sets have NN in their x-axis label. The dots show individual cluster lengths.

The intergenic clustering results identify large genes which may be missed in a cluster agglomerative clustering. The distribution of the gene lengths in all four sets of clusters is shown in Figure 4.3. The intergenic clustering resulted in finding longer genes in cluster compared to agglomerative clustering, but the differences were not statistically significant. The median gene length between

protein-coding clusters created with agglomerative clustering is about ˜7 kbp and is ˜11 kbp using intergenic clustering. The median gene lengths between clusters made using disease genes are ˜9.8 kbp for agglomerative clustering and ˜10.9 kbp for intergenic clustering.

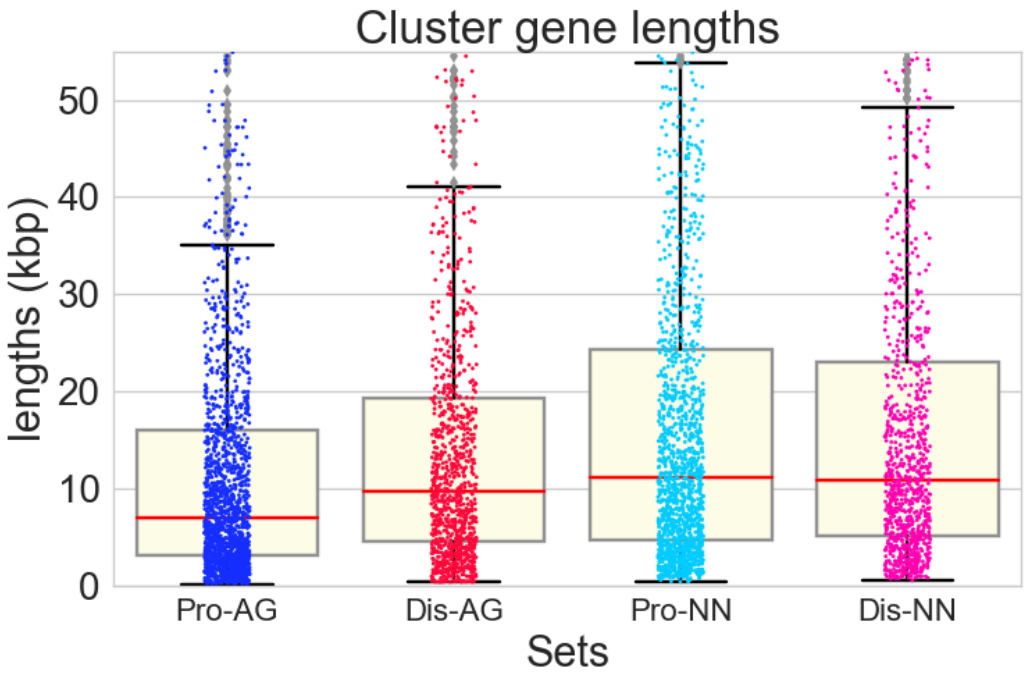

Figure 4.3: **Gene-length distribution in the four cluster sets.** Shown are the distributions of gene lengths in all four cluster sets. Protein-coding clusters are labeled on the x-axis with Pro. Disease clusters are names using Dis. Agglomerative clustering is labeled on the x-axis with AG. Intergenic clustering is labeled using NN.

Most of the longest genes are not found in clusters of ten or more genes (Figure 4.4, Pr-All): Intergenic clustering using a maximum distance of 10 kb between protein-coding genes and 50 kb between disease genes and only keeping clusters with ten or more genes results in a median gene length for all

˜20k protein-coding genes of 24 kb and 11kb for the protein-coding genes found in clusters.

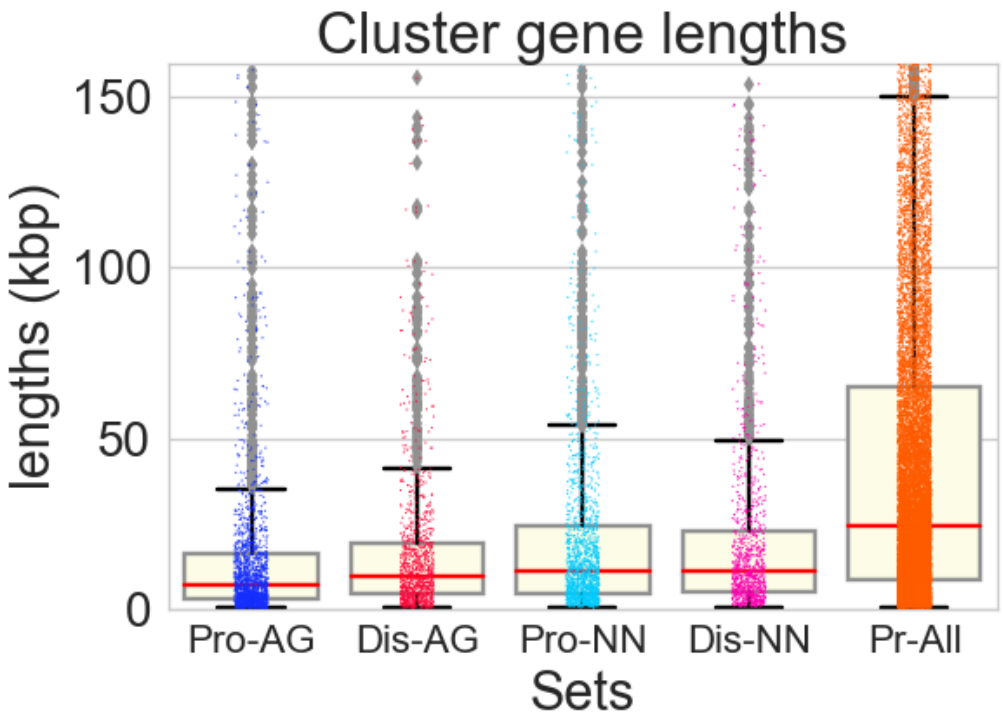

Figure 4.4: **Gene-length distributions in clusters compared to all protein-coding genes.** The gene length data from Figure 4.3 are shown next to the boxplot showing the distribution of gene lengths for all ˜20k protein-coding genes in the human genome, shown on the right with each orange dot representing the length of one gene.

Protein-coding clusters and disease clusters often did not overlap using the hierarchical agglomerative clustering but did often overlap when using the intergenic clustering. The percentages of disease genes in each protein-coding gene clusters are shown in Figure 4.5. A cyan on the leftmost side of the figure indicates that 8% of the protein-coding clusters found using agglomerative

clustering have no disease genes. The rightmost red bar indicates that only 1% of all protein-coding clusters contain genes where 90 to 95% of the genes are associated with disease. Most protein-coding clusters contained genes where between 10% and 75% are associated with disease. The protein-coding clusters with no disease association generally contained genes which were all in the same gene family per cluster. An example of a gene cluster with no disease association is the 17a21.2 cluster containing keratin-associated proteins such as KRTAP1-1, KRTAP1-3, KRTAP1-4, etc.

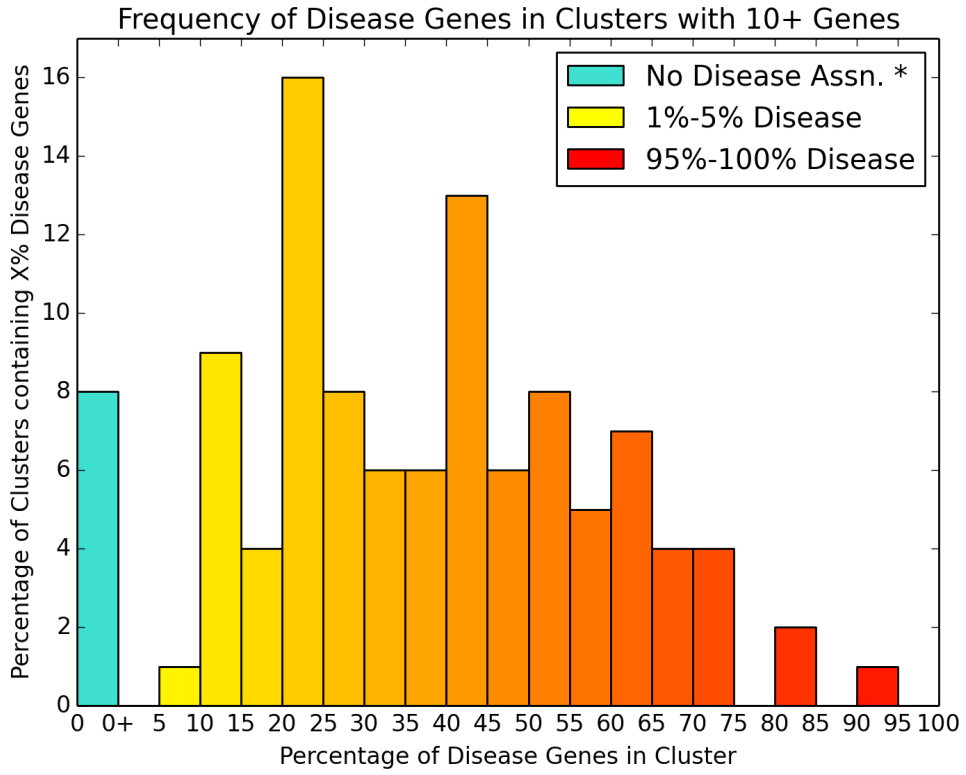

Figure 4.5: **Clusters of protein-coding genes and their disease-associated gene content.** The cyan bar indicates that 8% of all clusters of protein-coding genes contain no disease-associated genes. The red bar on the right indicates that 1% of all protein-coding clusters are composed of 90-95% disease-associated genes.

Of the top 50 most disease-associated genes, only four (IL4, CD14, TNF, and AGER) were found in the set of protein-coding clusters containing 10 or more genes. Thirty of the 50 were found in protein-coding clusters containing any number of genes rather than only the 10+ gene cutoff used to identify clusters for focused study. Twenty of the top 50 disease-associated genes were not in any clusters, even the smaller clusters. Examples of highly associated disease genes not within 50k bps of any protein-coding gene include C-reactive

Protein (CRP), Prostaglandin-Endoperoxide Synthase 2 (PTGS2), C-X-C motif chemokine Ligand 8 (CXCL8), Vascular Endothelial Growth Factor A (VEGFA), Interleukin 6 (IL6), Toll Like Receptor 4 (TLR4), and Nitric Oxide Synthase 2 (NOS2).

The relative disease association for disease clusters in the human genome is shown in Figure 4.6. The disease types shown are cancer (C), infection (I), autoimmune (A), heart (H), neurological (N), and environmental (E). The letter height represents the percentage of genes in the cluster associated with any of the six disease types. Runs of diseases where neighboring genes are all associated with a single disease can appear as a large letter. Letters may also appear large if many different diseases in the same class appear in the cluster; however, runs of diseases may also appear as a shorter letter if many other diseases of another disease type are also in the cluster.

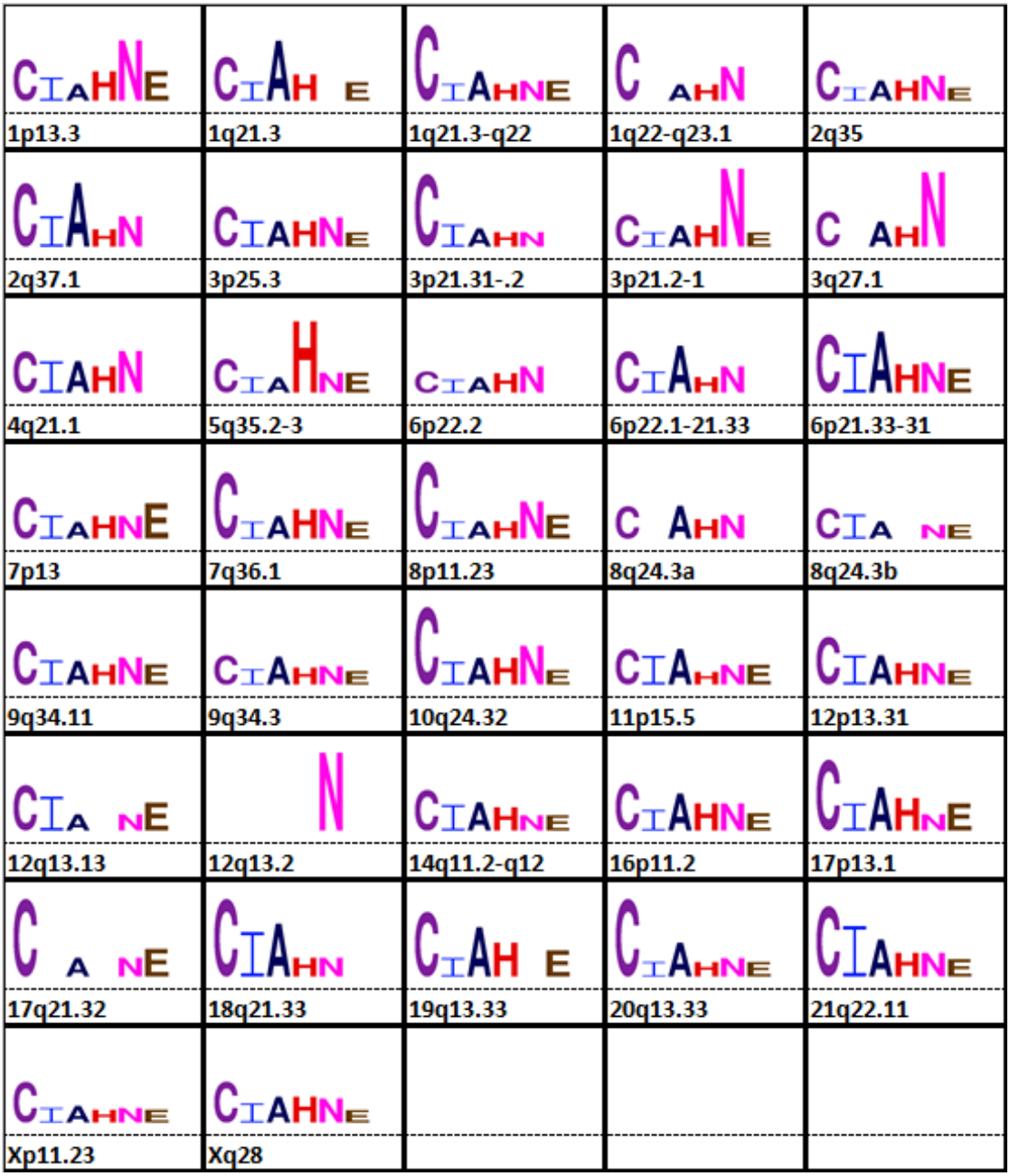

Figure 4.6: **Disease-class content in disease clusters.** Each box indicates one disease cluster and has a corresponding cluster in the genome plot in part b. The relative content of disease genes in a specific disease class is indicated by the colored letters. For example, the disease cluster at 12q13.2 contains disease genes only for neurological diseases.

4.8 Disease in clusters

Two visualizations help summarize a cluster, where one uses words from the abstract of publications about the cluster while another uses ASCII art to compare the disease associations of the genes within a cluster and to compare clusters. The ASCII Art visualization is described in the next section.

The first search yielded thousands of publications per cluster, and a second search for each disease using NCBI's PubMed also resulted in many publications. Intersecting both sets of publications for each cluster often reduced the results from thousands to hundreds of publications per cluster, resulting in the final list of publications related to genes in a cluster.

The first word cloud for a cluster used words found in the title and abstract of the publications of interest, while the second used MeSH terms associated with each set of publications. MeSH terms are Medical Subject Headings from the United States National Library of Medicine's controller hierarchical vocabulary. The hierarchy of the MeSH terms allows searching for broad or specific headings, and most publications in PubMed have an attached set of MeSH terms.

Figure 4.7 shows an example of the two sets of word clouds for a cluster on chromosome 6, which contains numerous HLA (Human Leukocyte Antigen) genes that are highly associated with disease.

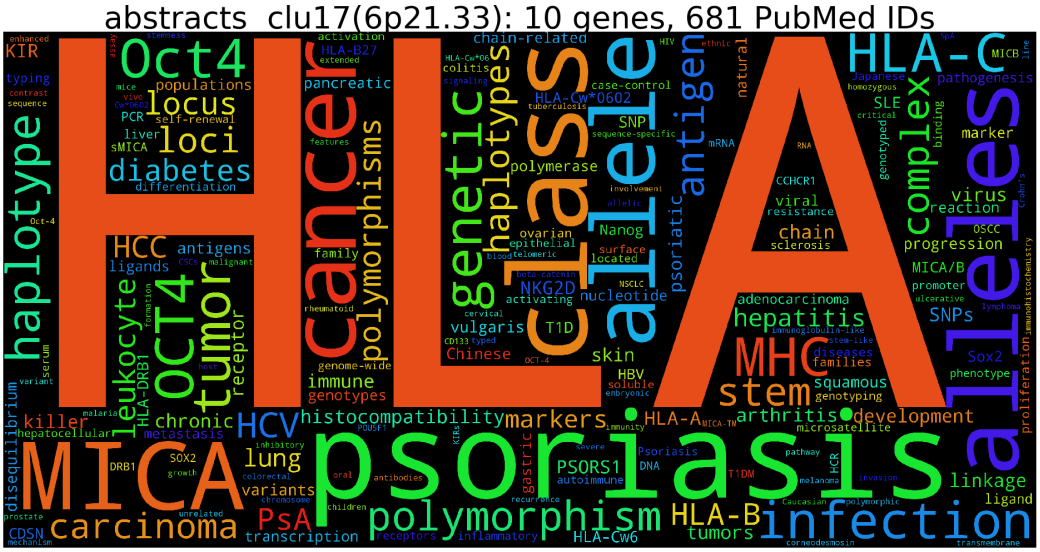

Figure 4.8: **Word cloud for chr6 HLA cluster using MeSH terms.**

4.8.1 Visualizing clusters

Comparing over 100 clusters necessitated an efficient and information-rich method to simultaneously visualize multiple clusters. Using ASCII art enables comparing many genes, annotated with disease association and the ranking of well-studied vs. understudied, as described below.

Figure 4.9 shows the six general disease classes used in this thesis: Cancer, Nervous System Disorder, Autoimmune, Heart, Infection, and Environmental. The 40+ diseases in this thesis are grouped into these 6 disease classes. The "Key" in Figure 4.9 is the ASCII letter used in the ASCII art visualization to represent a disease class.

Disease Class	Key	Genes
Cancer	C	7133
Nervous System Disorder	N	3101
Autoimmune	A	2839
Heart	H	2290
infection	I	1871
Environmental	E	1667

Figure 4.9: **Disease classes and gene counts.** Individual diseases are grouped into six disease classes. Each class has a "Key", which is a single letter (C, N, A, H, I, E) used to identify the class. The number of genes for each disease class appears in the "Genes" column. There is one color per disease class: Cancer: purple; Nervous System Disorder: fuchsia; Autoimmune: dark blue; Heart; red; Environmental: brown;

Figure 4.10 shows the ASCII key for the individual diseases as well as the gene counts for each disease for the organisms of human, mouse, and fly. The gene counts significantly differ across the species despite each search being

identical except for the species name. This might result from different experiments conducted in each of the species; for example, human cells may be used in experiments, while the entire mouse or fly can be used in other experiments.

Disease	Key	hsa	mmu	dme	Disease	Key	hsa	mmu	dme
Lung Cancer	Ca	2459	226	1	Rheumatoid Arthritis	Aa	1097	141	1
Prostate Cancer	Cb	2413	287	0	Asthma	Ab	849	351	0
Hepatocellular Cancer	Cc	2248	261	1	Lupus	Ac	824	252	1
Colorectal Cancer	Cd	2230	187	1	Multiple Sclerosis	Ad	777	118	0
Gastric Cancer	Ce	1995	104	0	Crohns Disease	Ae	557	29	0
Squamous Cell Cancer	Cf	1897	115	4	Psoriasis	Af	535	117	1
Ovarian Cancer	Cg	1611	90	1	Ulcerative Colitis	Ag	528	56	0
Melanoma	Ch	1548	490	1	Pancreatitis	Ah	281	184	0
Glioma	Ci	1414	145	1	Diabetes Type I	Ai	3	0	0
Pancreatic Cancer	Cj	1270	126	0	Hypertension	Ha	921	434	1
Lymphoma	Ck	1144	312	9	Atherosclerosis	Hb	844	652	1
Renal Cell Carcinoma	Cl	891	39	0	Coronary Artery Dis.	Hc	732	12	0
Breast Invasive Cancer	Cm	6	0	0	Stroke	Hd	712	292	1
Schizophrenia	Na	1635	127	3	Heart Failure	He	590	315	2
Autism	Nb	549	103	10	Heart Disease	Hf	556	59	1
Alzheimer	Nc	519	148	3	Hepatitis C	Ia	1025	107	30
Parkinsons Disease	Nd	477	161	15	Hepatitis B	Ib	811	90	0
ALS	Ne	404	138	7	Tuberculosis	Ic	430	198	0
Mental Retardation	Nf	357	38	9	Influenza	Id	355	236	1
Major Depression	Ng	351	13	0	Malaria	Ie	206	130	1
Charcot Marie Tooth	Nh	97	25	2	Staphylococcus	If	90	67	3
Spinal Muscular Atroph	Ni	62	37	2	Obesity	Ea	1667	859	11

Figure 4.10: **Disease gene counts for human, mouse, and fly.** The "Disease" column contains the name of an individual disease in this two-column table. The gene counts for human, mouse, and fly are identified under the columns, "hsa", "mmu", and "dme" respectively. The key for each disease is a two-letter combination. The first letter is the disease class, while the second letter represents the individual disease. There is one color per disease class, which are defined in the previous table.

The major histocompatibility complex region on chromosome 6, a cluster circled in red in Figure 4.11, is involved in nearly all the diseases in this study. The run of genes circled in Figure 4.11 and their association with disease are shown in Figure 4.12.

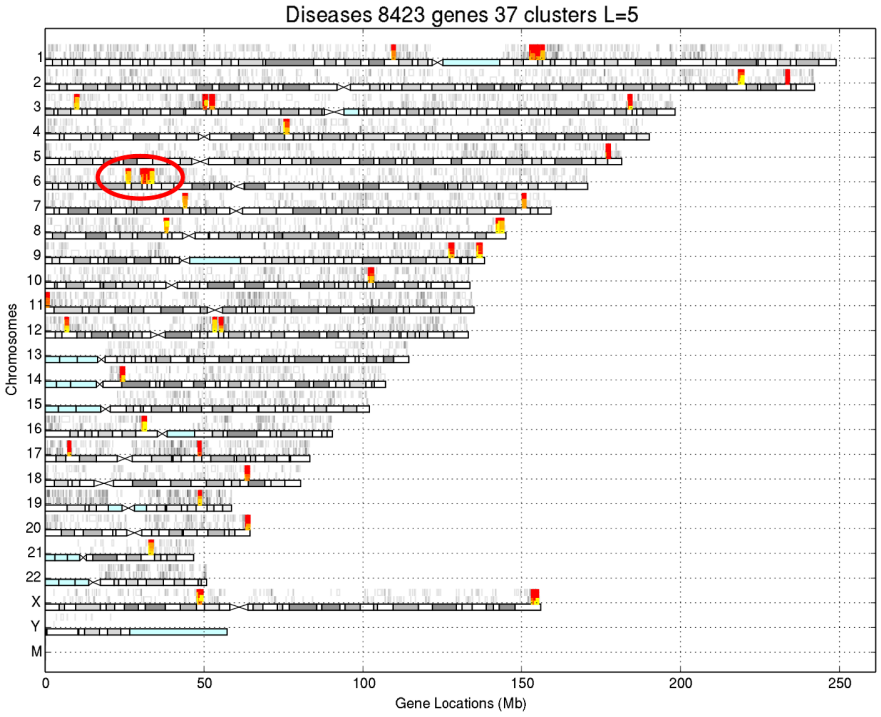

Figure 4.11: **Cluster of genes associated with disease.** The twenty-three chromosomes of the human genomes appear as ideograms with the boxes of each chromosomal region colored white through gray and light blue. The centromeres are denoted in each chromosome at the narrowest part. Genes are shown above each chromosome in two rows. The first row (light gray lines) are the genes in the forward strand, while the second row are in the reverse strand. Each cluster of genes associated with disease is highlighted with red, orange, and yellow. Genes that are tightly clustered are red. Genes loosely clustered are yellow. Genes clustered between tight and loose are orange. The cluster circled in red is the histocompatibility complex region on chromosome 6.

The genes are in genomic order in Figure 4.12. There is one row per gene, with the gene symbol on the rightmost side. The IDG column on the left indicates how well studied a gene is.

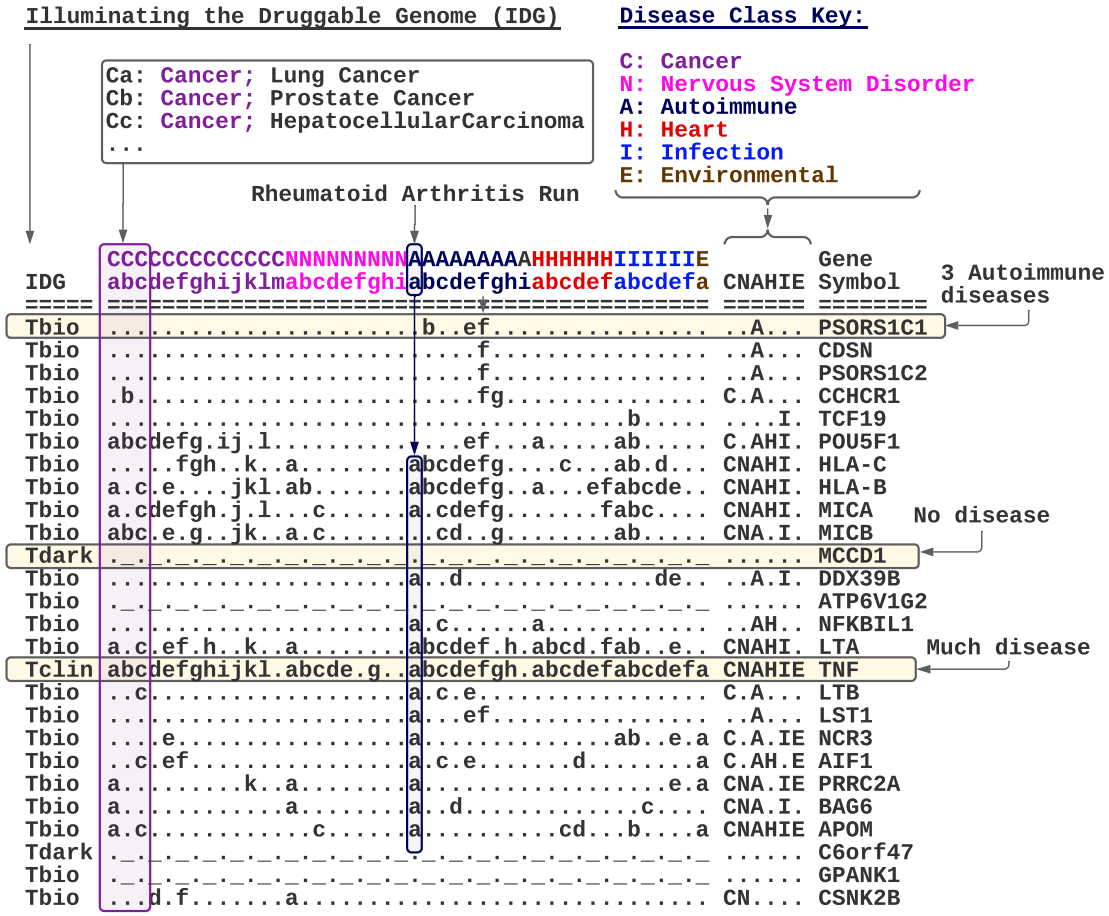

Figure 4.12: **ASCII art summarizing one cluster.** Each row represents one gene, whose symbol is in the rightmost column. The first column is the IDG ranking, with Tclin being the best studied gene and Tdark being the least studied gene. The second set of columns marks the individual disease keys (Figure 4.10). Each single letter (a, b, c, etc.) indicates that the disease is associated with the gene. A "." indicates that the disease is not associated with the gene. A dot-dash pattern indicates that the gene is not associated with any diseases. The third set of columns is a summary for the disease class (Figure 4.9).

tumor necrosis factor (TNF)

The gene TNF, highlighted in light yellow, is the most studied gene and is marked as Tclin, the most studied IDG ranking. TNF is associated with the most diseases of any gene studied in this thesis.

psoriasis susceptibility 1 candidate 1 (PSORS1C1)

The second set of columns, the widest set block, shows the individual diseases associated with a single gene; for example, the PSORS1C1 gene, highlighted in yellow at the top gene row, is associated with only three autoimmune diseases and marked as "b", "e," and "f" under the "AAAAAAAAA" (autoimmune) top header line. The marks "b", "e," and "f" are the autoimmune diseases asthma, Crohn's disease, and Psoriasis, as defined in Figure 4.10.

The third set of columns, headed with "CNAHIE," enables viewing which disease classes are associated with this gene. The "A" in the yellow highlighted "..A..." for PSORS1C1 indicates that all diseases in this study that are associated with PSORS1C1 are autoimmune diseases (Figure 4.9). The "."s under the C, N, H, I, E headers indicate that there are no diseases associated with PSORS1C1 in the classes: Cancer, Nervous System Disorder, Heart, Infection, and Environmental.

The IDG marking of PSORS1C1 is "Tbio," indicating that the gene is understudied.

mitochondrial coiled-coil domain 1 (MCCD2)

The MCCD2 gene is not associated with any diseases in this study and is marked with a dot-dash-dot-dash pattern in the individual diseases column to be quickly distinguishable from genes with disease associations. The IDG marking is "Tdark," which indicates that this gene is one of the least studied.

4.8.2 General impressions

Genes that are highly associated with disease tend to not cluster together, but instead a high disease gene is frequently flanked by genes associated with little or no disease. The cluster shown in Figure 4.12 is unusual since it has several other genes that are highly annotated with disease, including HLA-B, LTA, and MICA, which are respectively "major histocompatibility complex, class I, B", "lymphotoxin alpha," and "MHC class I polypeptide-related sequence A."

The diseases associated with genes tend to be sprinkled about the cluster, but with some diseases, runs of consecutive genes are associated with one disease. In the example in Figure 4.12, a run of genes is associated with Rheumatoid arthritis ("Aa"). The Rheumatoid arthritis gene run in Figure 4.12 contains multiple gene family types and appears in several areas across the genome.

The understudied genes in the human genome, labeled as Tbio or Tdark, are associated with a maximum of 12 diseases in this thesis. The well-studied genes, labeled as Tclin or Tchem, can be associated with over 40 diseases.

All genes of all IDG types (Tclin, Tchem, Tbio, and Tdark), however, are skewed to be highly associated with few or even no diseases in this study.

Tclin genes associated with few diseases may be due to the drugs targeting that gene's products only being used for a small number of diseases. Tclin genes associated with no diseases in this study may be due to annotations not yet made or because the gene targets a disease outside this study.

Chapter 5: Grouping genes by function

5.1 Prelude

The Gene Ontology Enrichment Analysis (GOEA) results for each cluster in the genome are critical to this thesis. There were two motivations for my work on the GOA TOOLs paper: The first was verifying that the GOEA results were correct and complete, while the second was to find a method to better summarize the GOEA results.

5.1.1 Open questions regarding GOEAs

For the first motivation, there were four major open questions about GOEA not covered in the literature: How correct are the Gene Ontology (GO) results?; How many expected GO results are missing?; What affects the number of results that are missing? Should the optional argument, "propagate counts" be used? I designed and created 100,000 stochastic GOEA simulations to answer these questions.

How correct are the Gene Ontology (GO) results?

The False Discovery Rate (FDR) used in my GOEAs was set to 0.05, meaning that at most, 5% of the results may be false positives. Is a maximum of 5% false positives accurate? It turned out, "No, it is not." False positive percentages were well over 5%, reaching a maximum of 18%, as shown in the red text in Figure 5.1.

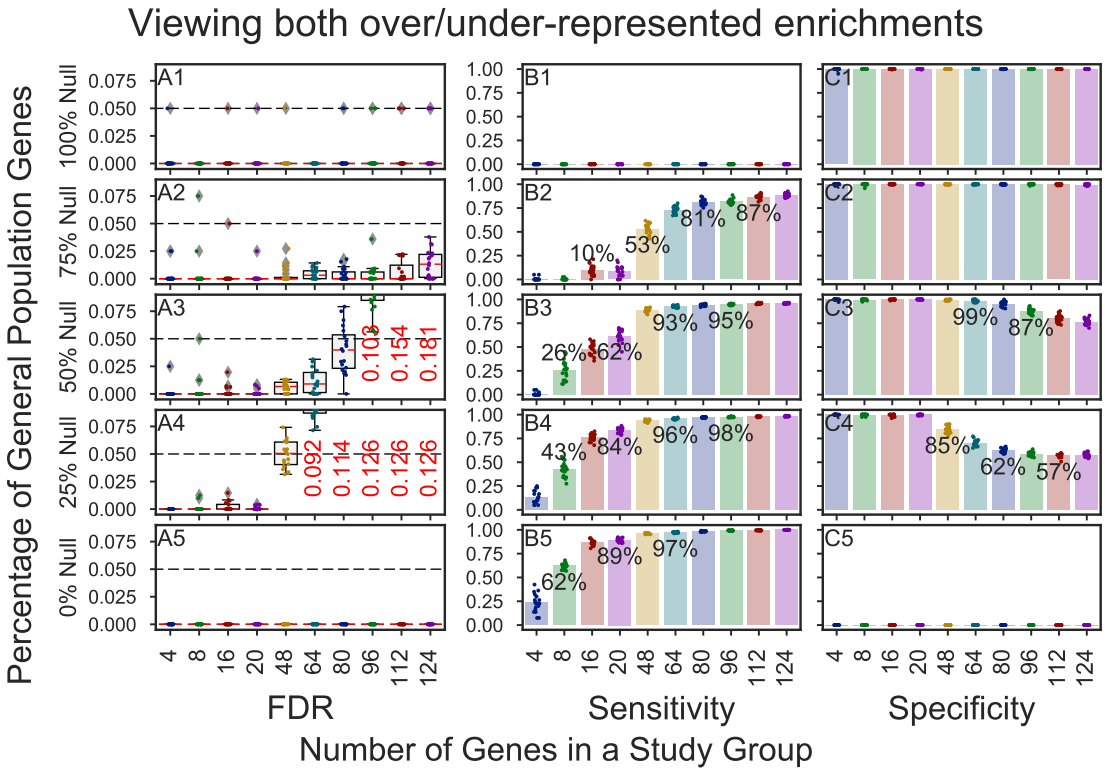

Figure 5.1: **The first GOA TOOLs GOEA simulations fail in panels A3 and A4 with FDR values exceeding the alpha of 0.05 set by the researcher.** The values of failing FDRs are shown using red text. False Discovery Rates could be as high as 18%, as seen in the more-left red test in panel A3.

Viewing both the over-enriched (enriched) and under-enriched (purified) results caused this exceedingly high set of false positives. When only viewing the over-enriched results, the FDR is drastically reduced, as shown in Figure 5.2; however there are failure rates above an FDR of 0.5 (the dotted line in the A panels), as shown in panels A2 and A3. Further investigation shows that the source of these failures is the inclusion of extremely broad GO terms. The median and mean number of genes associated with each GO term are 3 and 16 genes respectively, with a standard deviation of the mean being 128. The

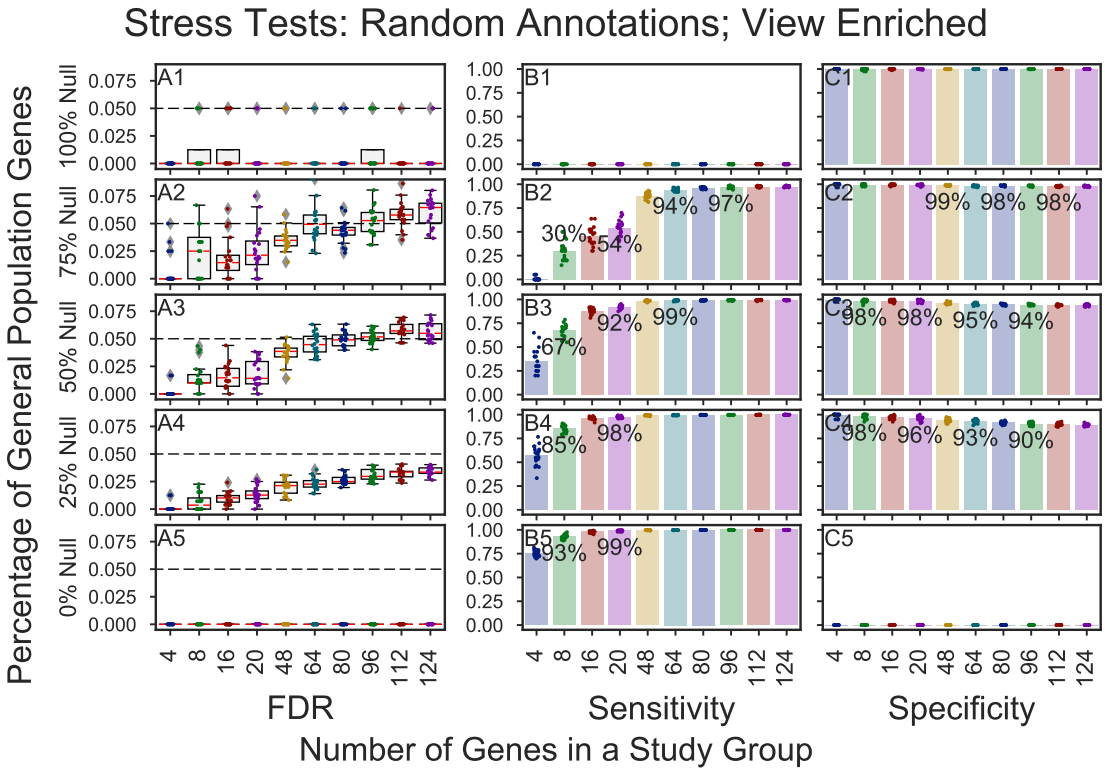

Figure 5.2: **GOA TOOLs GOEAs stress tests with randomly shuffled associations nearly pass if only enriched GO terms are viewed.** The associations are randomly shuffled while still maintaining the distribution number of GO terms per gene. The failing FDRs (above 0.05) are shown in panels A2 and A3 for gene groups having 96, 112, or 124 genes.

number of genes associated with each GO term ranged from 1 to 7,000, with almost all genes annotated by a small number of GO terms. When removing GO terms annotated by 1,000 or more genes, which accounts for about 30 GO terms out of 47,000, the FDR stays under 0.05 as shown by all box plots in the A panels being under the dotted FDR line in Figure 5.3. When these genes were removed, the false positive rate stayed under the research-specified FDR of 0.05.

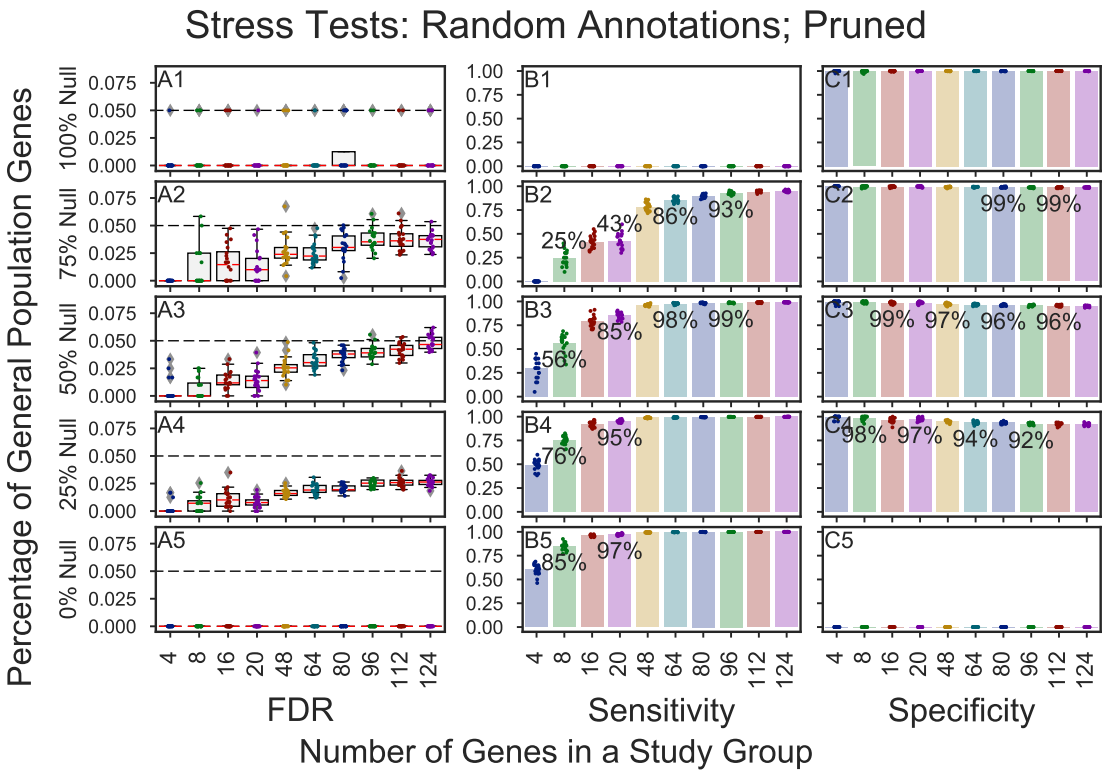

Figure 5.3: **GOA TOOLs GOEAs stress tests with randomly shuffled associations pass for all cases if only 30 out of over 17k+ GO terms associated with more than 1,000 genes are removed.** The median number of genes per GO term in the mouse associations is 3 genes/GO. Genes per GO term range from 1 gene to ~7k genes per GO term. (mean=16 genes/GO, SD=128).

Investigating missing GOEA results: false negatives

To answer the questions "How many expected GO results are missing?" and "What affects the number of results that are missing?" I created two sets of 10,000 GOEA simulations each, randomly choosing from a target set and background set of genes for each simulation. The first set, the small target set, contains 124 genes annotated with GO terms related to humoral response. The

second set, the large background set, is the set of all genes annotated with GO terms that are not related to humoral response.

The results appear in Figure 5.4, part A. The answer to the question, "How many expected GO results are missing?" is provided by examining the white space above the bars on panels B2 through B5. More white space above the bars means humoral response genes were not identified as such. For example, in panel B2, no genes are found to be enriched for humoral response for gene set sizes of 4, when there is, in fact, a humoral response gene present. In panel B2, genes truly enriched in humoral response are only identified 10% of the time for gene sets of 20 genes, where 5 genes are humoral response genes. The panels B2 to B5 show a large percentage of missing genes, especially for small sets of genes (4, 8, 16, and 20 genes). Because my thesis involves gene clusters with 10 of more genes, GOEAs with missing expected enriched results will affect the results, which was not known until performing the simulations and analyzing the results.

The answer to the question, "What affects the number of results that are missing?" is in two parts: The first part is "gene set size affects the numbers of false negatives" and follows from showing that bigger gene sets produce better results, with the results becoming quite poor if the gene set size is under twenty genes. The second part of the answer is "gene set composition affects the numbers of false negatives" and can be seen by examining how B2, B3, B4, and B5 differ. Gene set composition is the percentage of genes in a gene set that are humoral response genes. In B2, 25% of the genes in each gene set are humoral response genes, which is labeled as 75% Null on the y-axis. In B3, 50% of the genes are humoral response genes (50% Null), and in B5, all the genes

are humoral response genes (0% Null). Better results are returned for gene sets with higher percentages of actually enriched genes. For example, in the 20-gene set size, 90% of the expected humoral response genes will be identified to be enriched in humoral response if all the 20 genes are humoral genes (Figure 5.4, panel B5), but if only 5 genes in the 20 gene set are enriched in humoral response, they will be identified only 10% of the time (Figure 5.4, panel B2). The percentage of actually enriched genes in my clusters cannot be controlled for this thesis, but the information concerning missing expected results is helpful. Next, I investigated how to improve the GOEA results for small gene sets.

Should the optional argument "propagate counts" be used?

If the "propagate counts" argument is set to false, the annotations of genes to GO terms are used with no modifications. A leaf-level GO term will often be annotated with a gene, but the next couple ancestors above the gene will have no annotations. The argument "propagate counts" causes any annotation on a lower-level GO term to be added to all GO term ancestors above it.

My concern was that using "propagate counts" would lead to too many false positives. Upon comparing simulations with "propagate counts" set to false with those set to true, I found that the enrichment analysis results were substantially better, with drastically fewer targeted genes missed if "propagate counts" was set to true (Figure 5.6 part B). Nearly all humoral response genes were returned except for the 4 gene set with 50% humoral response genes (Figure 5.6, panel B3) and the 4, 8, and 16 gene sets with 25% humoral response genes (Figure 5.6, panel B2). Because "propagate counts" causes terms at the

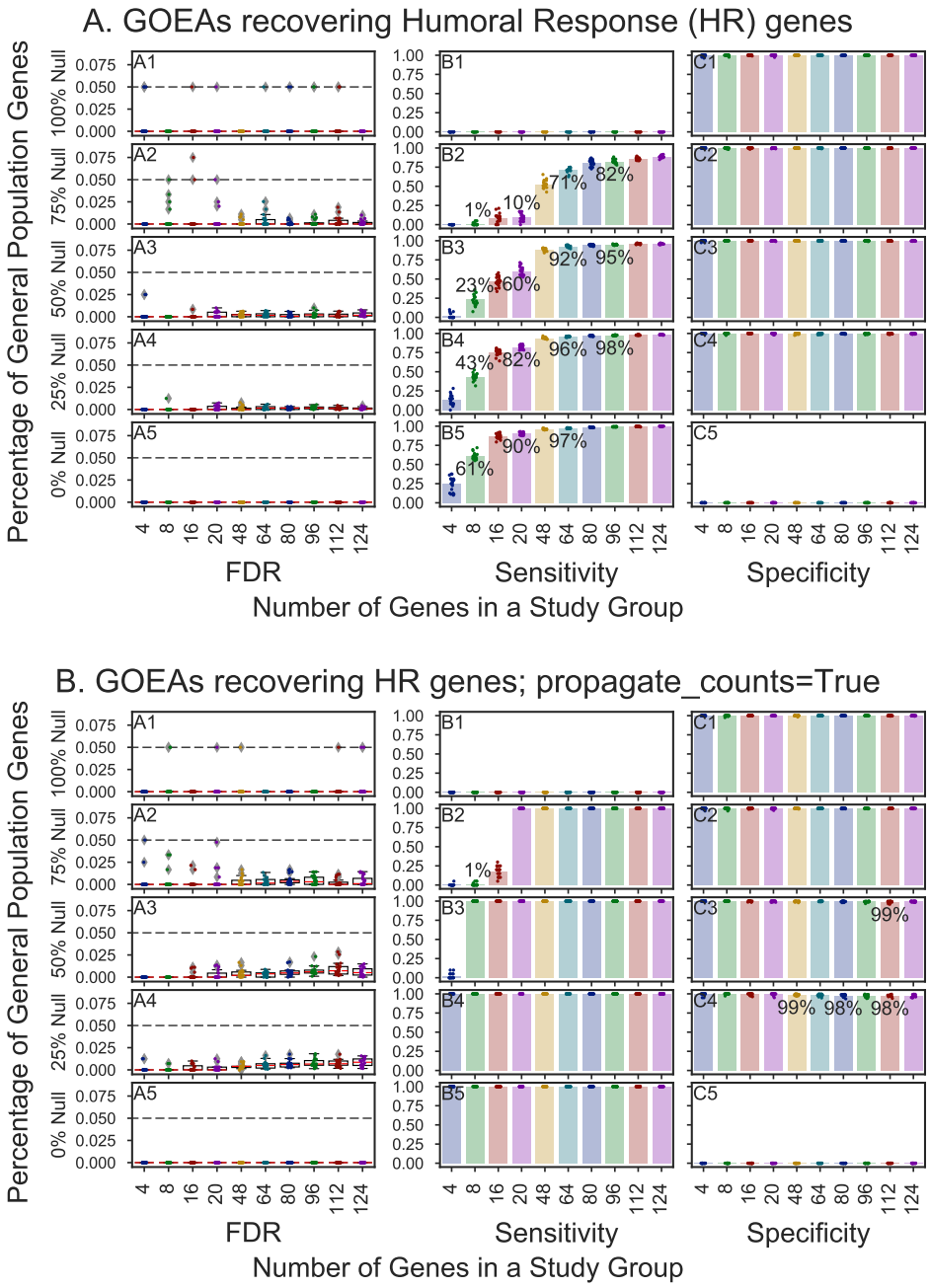

Figure 5.4: **Results for 20,000 GOA TOOLs GOEA stochastic simulations with varying sensitivity and consistently high specificity.** GOEAs performed well on study groups of 8+ genes if the GOA TOOLs GOEA option propagate_counts set to True.

top of the GO DAG, near the root term, to be annotated with thousands of genes, it is important to remove the broadest GO terms that add no new information to the analysis.

Writing on and publishing this paper in the journal Scientific Reports resulted in three conclusions regarding GOEAs: First, "propagate counts" should be set to true; second, the broadest 30 GO terms should be removed from the enrichment analysis; third, the enriched results should be viewed separately from the purified results. Finally, there is a risk of not finding truly enriched results, but this occurs when there are few target genes in the set and the gene set size is small.

5.1.2 Summarizing a set of GO terms

Summarizing a list of GO terms could be difficult, even for a small group of GO terms like the one in Table 5.1 part A, which is sorted by P-value, a common method of listing GO terms. Sorting by P-value is not helpful overall, unless there is a large magnitude of difference in the P-values. One GO term with slightly larger P-values than the next GO term down the list does not mean the first GO term is more important. Both GO terms are equally important if their P-value difference is small, and thus sorting by P-value does not help organize the list. Part A of Table 5.1 is challenging to summarize, even with a list as small as 12 GO terms.

My new method for grouping uses higher-level GO terms to group the lower-level GO terms. Although this is the common starting point for all GO grouping methods, my method differs by enabling the researcher to make grouping choices when there are two equally valid upper-level GO terms,

which occurs because GO terms frequently have multiple parent terms. The researcher will choose which upper-level GO term best focuses on their current research question.

Other grouping methods remove GO terms to make the list easier to read. The concern is that the researcher lacks the power to influence which GO terms are removed. Other grouping methods display the grouping as a figure. I found the list format to be easier to use, especially when working with the list in a text format on the command-line.

My method begins with the GO slims, a reduced set of broad GO terms provided by the Gene Ontology Consortium, which allows easily adjusting this list to include more precise GO terms. The researcher creates their own section names, such as "cell death" and "immune," which hold a number of GO slims. When the researcher moves a GO slim into the section, all terms below that GO slim are swept into the same section.

Using my novel method, the list of GO terms in Table 5.1 part A become grouped into four larger groups that encompass many GO terms: "cell death," "immune," "signaling," and "viral/bacteria," as shown in Table 5.1 part B. With the GO terms now grouped, the list can be summarized as: "Virus and bacteria are seen during an immune response, where signaling and cell death are enriched." Normally GO grouping lists are black and white, but a table can be colorized (shown in Table 5.1) if provided with a Python dictionary where the key is the section and the value is the color. The creation of section names for grouping is described in detail in the text.

A) Ungrouped GO IDs sorted by P-value

GO Name	P-value
immune system process	3.74E-07
defense response to protozoan	5.56E-06
defense response to virus	2.93E-04
+reg. of extrinsic apoptotic signaling pathway	5.94E-04
positive regulation of T cell mediated cytotoxicity	7.30E-04
response to bacterium	7.30E-04
+reg. of cysteine-type endopeptidase in apoptotic process	1.35E-02
pyroptosis	1.86E-02
+reg. of I-kappaB kinase/NF-kappaB signaling	3.15E-02
+reg. of TNF-mediated signaling pathway	3.70E-02
antigen processing and presentation of exogenous antigen	4.32E-02
purinergic nucleotide receptor signaling pathway	4.32E-02

B) Grouped GO IDs sorted by GO group, then P-value

GO Name	P-value	Key	Section
+reg. of extrinsic apoptotic signaling pathway	5.94E-04	X	cell death
+reg. of cysteine-type endopeptidase in apoptotic process	1.35E-02	X	cell death
pyroptosis	1.86E-02	X	cell death
immune system process	3.74E-07	C	immune
+reg. of T cell mediated cytotoxicity	7.30E-04	C	immune
processing/presentation of exogenous antigen	4.32E-02	C	immune
+reg. of I-kappaB kinase/NF-kappaB signaling	3.15E-02	S	signaling
+reg. of TNF-mediated signaling pathway	3.70E-02	S	signaling
purinergic nucleotide receptor pathway	4.32E-02	S	signaling
defense response to protozoan	5.56E-06	H	viral/bacteria
defense response to virus	2.93E-04	H	viral/bacteria
response to bacterium	7.30E-04	H	viral/bacteria

Table 5.1: **GOA TOOLs grouping makes GO lists easier to read.** GO terms in Table A are sorted by P-value. In Table B, GO terms are grouped first and then sorted by P-value. These tables were produced using GOA TOOLs grouping and table writing code. The first column, 'GO Name' is the name of the GO term found in the GO DAG. The second column shows the P-value obtained from running a GOEA analysis. The third column shows the functional group containing the GO term, called a 'section name.' +reg: positive regulation; NF: nuclear factor; TNF: tumor necrosis factor

5.2 GOA TOOLs Introduction

The biological interpretation of gene lists with interesting shared properties, such as up- or down-regulation in a particular experiment, is typically

accomplished using gene ontology enrichment analysis tools. Given a list of genes, a gene ontology (GO) enrichment analysis may return hundreds of statistically significant GO results in a "flat" list, which can be challenging to summarize. It can also be difficult to keep pace with rapidly expanding biological knowledge, which often results in daily changes to any of the over 47,000 gene ontologies that describe biological knowledge. GOATOOLS, a Python-based library, makes it more efficient to stay current with the latest ontologies and annotations, perform gene ontology enrichment analyses to determine over- and under-represented terms, and organize results for greater clarity and easier interpretation using a novel GOATOOLS GO grouping method. I performed functional analyses on both stochastic simulation data and real data from a published RNA-seq study to compare the enrichment results from GOATOOLS to two other popular tools: DAVID and GOstats. GOATOOLS is freely available through GitHub: `https://github.com/tanghaibao/goatools`.

Gene ontology enrichment analysis (GOEA) is used to test the overrepresentation of gene ontology terms in a list of genes or gene products in order to understand their biological significance. Members of the Gene Ontology Consortium (GOC) [11] from all over the world collaborate to develop the Gene Ontology (GO), a resource to describe the molecular function, cellular localization, and biological processes of gene products across multiple species. The ontology includes over 47,000 terms (as of April 2018) and describes formal relationships among them. GOC members annotate ontology terms to specific gene products on the basis of experimental and computational prediction [33]. Annotation coverage of GO terms to individual

genes is high for humans and model organisms, with 85% of ˜20k human protein-coding genes having GO annotations, 90% of the ˜22k Ensembl mouse genes, and 77% of the ˜14k fly genes. Both ontologies and annotations can change incrementally on a daily basis [36]. To keep a laboratory's many functional genomic studies up-to-date with the rapidly evolving biological knowledge, it can be helpful to use a programmatic API built directly into an analysis pipeline; GOATOOLS does just that.

Python has a large, diverse open-source development community and comprehensive scientific computing libraries for building robust and reproducible computational workflows. GOATOOLS allows GO term manipulation, GOEA testing, and custom ontology visualization in gene functional studies.

I describe the GOATOOLS implementation first, followed by stochastic simulations, and finally demonstrate a case study using gene expression data from the paper by Gjoneska et al. (2015), *Conserved epigenomic signals in mice and humans reveal immune basis of Alzheimer's disease* [73]. Going forward, I will refer to this paper as the Gjoneska paper. I then compare GOATOOLS results with two mainstream methods: the web-based DAVID tool (Database for Annotation, Visualization, and Integrated Discovery) [90] and the R library, GOstats [57]. I demonstrate that GOATOOLS yields similar or better GOEA results and provides more flexibility via the Python API to group, sort, summarize, and visualize the results.

5.3 Materials and Methods

5.3.1 GOATOOLS development

GOATOOLS is open-source and available on GitHub (`https://github.com/tanghaibao/goatools`). In the tradition of open-source packages, GOATOOLS releases are early and often and have undergone continuous refinement over the last seven years. GOATOOLS was successfully used to explore a variety of research questions concerning a wide range of organisms, including over twenty different plant species, about ten fish species, five animal species, fungus, bacteria, and microalgae [32]. One publication that used and cited GOATOOLS investigated the immunogenetics of disease resistance of the common carp (*Cyprinus carpio*), an important aquacultured fish [117]. Another publication used GOATOOLS to study the maternal-to-zygotic transition of an embryo[14].

5.3.2 GOATOOLS implementation

GOATOOLS can be installed through package managers including Python *easy_install* or *pip*, and is also available as a bioconda package (`https://bioconda.github.io`). Extensive tutorials and Jupyter notebooks are available to demonstrate the usage of GOATOOLS in step-by-step fashion. In the following section, I detail the GOATOOLS implementation, including details on file I/O, data structure, statistical testing, reporting, and visualization.

5.3.3 File I/O and data structure

A GOEA requires both a copy of the ontology, which describes terms and relationships among them, and a set of annotations, which associates the GO terms to specific gene products. The ontology is available from the main GO website (`http://geneontology.org/page/download-ontology`). There are three versions of the GO ontology: GO-basic, GO, and GO-plus, among which only GO-basic is guaranteed to be acyclic [67]. GOATOOLS traverses the ontology, which is stored as a graph, and thus requires the acyclic version found in GO-basic; this is the GO version recommended for most GO-based annotation tools [67].

Most ontology systems are also becoming available in a JSON (JavaScript Object Notation) format, which is a lightweight, language independent interchange format [1]. The JSON file currently available is *go-plus.json*, which contains more extensive information than the smaller *go-basic.obo*. Accounting for the larger size of the more information-rich GO-plus file, the rate of reading and parsing the ontologies from the JSON file is about three times faster than the rate of reading the obo text file.

The annotations are currently available for download from the GOC as GO Annotation Format (GAF), from NCBI's FTP server in a gene2go format, or from the European Bioinformatics Institute's FTP site in the Gene Product Association Data format (GPAD). GOATOOLS can efficiently parse these relevant file formats to retrieve rich attributes of each term and model the term relationship using the *is_a* attribute as well as *part_of*, and *regulates* relationships into a directed acyclic graph (DAG) [23]. The DAG data structure allows

traversal of terms along the hierarchy for tasks such as determination of level or depth, retrieval of parent or child terms, and calculation of semantic similarities (e.g. Resnik's score [176] and Lin's score [121]) between terms. Mapping between regular GO terms and a restricted subset of GO (GO slims) is also supported.

GOATOOLS returns GOEA results in a variety of formats: EXCEL spreadsheet, tab-separated text file, JSON file, or Python variable containing a list of results with the GO results grouped by function as part of the API.

5.3.4 Statistical Testing

Many functional genomics studies look to see if any selected gene sets contain enrichment (or perhaps less common, under-representation) of certain functional classes, which is a critical goal in the study of differential gene regulation [11]. The frequency of genes for a particular GO term in the sample is compared to the frequency in the background. A P-value is then computed, often on the basis of Fisher's exact test [2]. Of the 68 GOEA tools reviewed by Huang et al., 20 support Fisher's exact test, which uses a hypergeometric distribution during the calculation. The raw hypergeometric test is also popular with 21 tools for determining uncorrected P-values. Tests seen in other tools include chi-square test, t-test, Z-score, and Kolmogorov-Smirnov test [89]. GOATOOLS currently uses the Fisher's exact test to compute uncorrected P-values. Many users preferred Fisher's exact test over, for example, the chi-square test because Fisher's exact test is more accurate [136]. Another popular tool, DAVID, uses Fisher's exact test, along with a modified EASE

score [90]. The review by Rivals et al. discusses trade-offs for various statistical tests specifically for testing the enrichment of GO terms [179].

Due to a large number of tests performed, the individual P-value should be corrected to control the false positive rates [152]. GOATOOLS contains a large collection of multiple test correction procedures (12 tests to date), which include all the functions available from the statsmodels Python library [189]. Each of these tests may be more appropriate when used under specific experimental settings or if able to offer different levels of stringency. I have implemented popular methods including Bonferroni, Sidak, and Holm, as well as False Discovery Rate (FDR) procedures such as Benjamini-Hochberg or resampling-based FDR [152].

As an example, to demonstrate why a researcher may want to choose one kind of multiple test correction over another, I consider two popular tests: Bonferroni, which controls the family wise error rate (FWER), and Benjamini/Hochberg, which controls the false discovery rate (FDR). The FWER is the probability that there will be at most one false positive. Thus, a FWER set at 0.05 means that there is a 5% chance that there will be even one false positive. The FDR quantifies the fraction of discoveries that are allowable as false positives. A FDR set to 0.05 means that I have accepted that up to 5% of my "statistically significant" results may be false positives.

The Bonferroni results are guaranteed to have fewer false positives than the FDR tests. But the drawback is that Bonferroni is extremely conservative, with the loss of statistical power resulting in many missed true positives. In other words, truly significant observations are discarded. FDR provides more true positive results overall, with the downside of more false positives up to a

maximum percentage of discoveries set by the researcher. FWER corrections like Bonferroni are desirable if a conclusion drawn from all ontology P-values for a set of genes would be invalidated if at least one of the P-values shows significance when there is none. Such strictness may not be desirable. For example, the conclusion, "a set of genes is rich in immune functions," is valid when many gene ontology tests correctly show significance for immune functions, but one test incorrectly shows significance for one specific immune function.

FDR controls have been recommended over Bonferroni-type multiple test corrections in health studies [74]. A recent paper by Goeman and Solari focuses on the trade-offs of the various multiple hypothesis tests [75]. The exhaustive list of statistical tests supported by GOATOOLS can be found at the GOATOOLS website (`https://github.com/tanghaibao/goatools#available-statistical-tests-for-calculating-uncorrected-P-values`).

5.3.5 Reporting

Gene Ontology Enrichment Analysis tools, when given a list of genes, can return hundreds of statistically significant GO results in a "flat" list, which can be challenging to summarize or to discern from a systems perspective using only a basic sort, like sorting all results by P-value. A "flat" list is a list of GO terms not organized with any consideration to the innate hierarchy that the GO terms have with one another [202].

The researcher may wish to retain all of the GOEA results, but re-organize them under general sections, like *immune* or *viral/bacteria*. In a flat list of GO terms sorted by P-value, GO terms related to interesting groups, like *immune*

or *viral/bacteria*, may be scattered throughout the list (Table 5.1A). Additionally, in a "flat" list it can be difficult to identify other general groups besides *immune* and *viral/bacteria* that might be present when the GO terms of various potential groups are interleaved among one another.

GOATOOLS grouping allows users to display GO terms and their associated study genes returned from GOEAs under general sections. GO terms in each section may then be sorted by P-value to easily see both the most statistically significant terms in *immune* and the most statistically significant GO terms in *viral/bacteria* (Table 5.1B).

The user can then reduce this list to produce a short summary list by printing only the top N sorted GO terms in each section, where N is a small number such as 1, 2, or 3.

5.3.6 Gene ontology graph layout

As of April 2018, the DAG contains over 47,000 GO terms and is divided into three major branches with each branch emanating from a single GO parent term at the top-level fanning out to over 28,000 of GO terms at the bottom level. The three broad top-level branch terms are *biological_process*, *molecular_function*, and *cellular_component*. GO terms may have more than one parent. There are over 20 GO children directly under the top-level branch term, *biological_process* (Table 5.2).

D1	dcnt	dep.	GO	name
	29,625	D00	GO:0008150	biological_process
A	18,703	D01	GO:0009987	cellular process
B	13,064	D01	GO:0065007	biological regulation
C	9,805	D01	GO:0008152	metabolic process
D	7,544	D01	GO:0032501	multicellular organismal process
E	6,473	D01	GO:0032502	developmental process
F	6,004	D01	GO:0050896	response to stimulus
G	4,354	D01	GO:0051179	localization
H	3,572	D01	GO:0071840	cellular component organization or biogenesis
I	2,369	D01	GO:0051704	multi-organism process
J	2,310	D01	GO:0023052	signaling
K	1,796	D01	GO:0002376	immune system process
L	1,277	D01	GO:0000003	reproduction
M	1,219	D01	GO:0022414	reproductive process
N	843	D01	GO:0040011	locomotion
O	492	D01	GO:0008283	cell proliferation
P	432	D01	GO:0040007	growth
Q	350	D01	GO:0022610	biological adhesion
R	280	D01	GO:0007610	behavior
S	113	D01	GO:0001906	cell killing
T	72	D01	GO:0044848	biological phase

Table 5.2: **The descendant counts of GO terms at depth-01 are highly skewed.** The root term, *biological_process* has over twenty GO children at depth-01 (D1) shown in the table sorted by their number of descendants (dcnt) with *cellular process* at the top having over 18,000 descendants and *cell killing* near the bottom having just over 100 descendants. The first column (D1 Alias) contains a letter used as an alias for each depth-01 GO term. The second column represents the total number of descendants from the specified GO term down to all of its leaf-level GO terms, which have no child GO terms. The third column, depth, shows the root term is at depth-00 and its children are at depth-01. The forth column, GO, is the ID for the term. The fifth column shows the human-readable name of the GO term. GO DAG relationships like *part_of* are used to count descendant counts in this table.

Letters like A, B, and C in the 'D1 Alias' column of Table 5.2 are aliases for depth-01 GO terms. The depth-01 aliases are used to provide the general location in the GO DAG of any one GO term. For example, in Table 5.2, 'Q' is the alias for *biological adhesion*. Immune GO terms descended from *biological adhesion* will have a 'Q' associated with them and include *lymphocyte aggregation* (Q), *positive regulation of gamma-delta T cell differentiation* (ABDEKQ), and *positive regulation of activated T cell proliferation* (ABKOQ). The aliases in the letters match the letters in Table 5.2 and are automatically created by the GOATOOLS code and included in the default printing format.

The number of descendants (descendant counts or *dcnt*) of each of the depth-01 GO children are dramatically skewed and have many shared parents. For example, as of 2018, the top term, biological_process, has over 29,000 descendant GO terms beneath it. The depth-01 GO term, *cellular_process*, just under the top term has more than 18,000 descendants while depth-01 *cell killing* has more than 100 descendants (Figure 5.5 and Table 5.2).

This illustrates that the descendant counts are highly skewed among all depth-01 terms. This sort of imbalance is seen throughout the DAG, not just at depth-01 (Table 5.3). Because of the highly skewed nature of the ontology graph, level or depth values cannot be used to estimate how close a GO term is to the bottom of the DAG [5].

Given a set of annotations, specificity of a GO term can be estimated from its association information content *tinfo* = –log (frequency), where frequency is the number of associations for the current GO term divided by the total number of associations in the full branch [126]. If no annotations are provided, using descendant counts (*dcnt*) under a GO term worked well in practice as an

GO Counts go-basic.obo Apr 4, 2018 47,216 Terms						
Depth	Depth			Level		
or Level	BP	MF	CC	BP	MF	CC
00	1	1	1	1	1	1
01	29	15	21	29	15	21
02	265	126	341	422	152	738
03	1270	527	499	2206	842	1077
04	2374	1515	732	4853	2068	1353
05	3697	4754	901	7310	4971	684
06	4468	1880	790	7215	1993	228
07	4687	1001	584	4658	776	63
08	4221	593	251	2003	207	10
09	3507	328	51	647	84	1
10	2401	160	4	244	13	0
11	1509	135	1	38	19	0
12	826	42	0	0	0	0
13	291	35	0	0	0	0
14	62	21	0	0	0	0
15	14	7	0	0	0	0
16	4	1	0	0	0	0

Table 5.3: **The counts of GO terms at all levels and depths is highly skewed across all three branches of the GO.** The deepest GO in the BP branch is at depth 16, while the deepest GO in the CC branch is depth 11. The GO roots are BP (*biological_process*), MF (*molecular_function*), and CC (*cellular_component*). The maximum length path from the root node down to the GO node is *Depth*. The minimum length path is *Level*.

estimate for defining how specific the GO term is, meaning how close it is to the bottom of the DAG. For example, it can be estimated that terms with thousands of descendants in the DAG, like *developmental process* with an information content (*tinfo*) of ˜5 and its over 6,000 descendants, are considered broader. Terms at the bottom of the DAG, like *germinal center formation* with a *tinfo* of almost 12 and having no descendants, are considered more specific

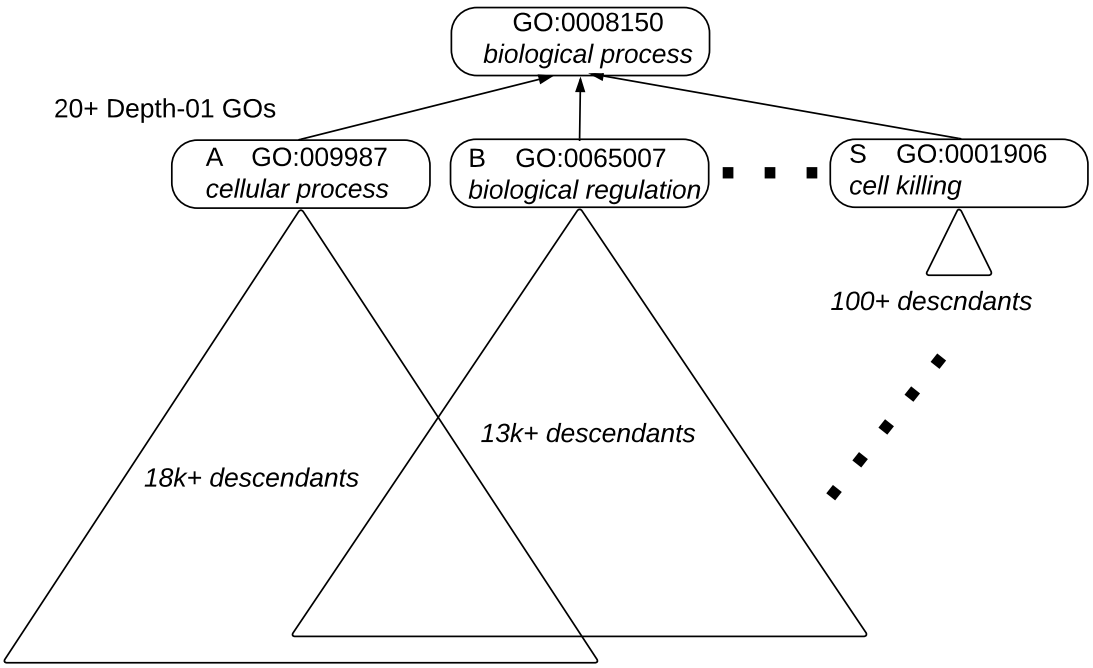

Figure 5.5: **The GO terms at depth-01 have highly skewed numbers of descendants from *cellular process* which has over 18,000 descendants down to *cell killing* having just over 100 descendants shown here.** GO terms within the overlapping triangles descend from both *cellular process* and *biological regulation*. The letters A, B, and S are aliases for the depth-01 GO terms as shown in Table 5.2. The ellipsis indicate that there are GO terms between *biological regulation* and *cell killing* that are omitted in the figure, but are shown in Table 5.2.

(Figure 5.6). GO terms with a descendant count of zero are at the bottom, or leaf-level of the DAG.

5.3.7 Grouping method

My novel approach to GO grouping retains all of the original GO IDs resulting from GOEAs, but rearranges the list so that the results are easier to read or print. The GOATOOLS method for grouping GO IDs uses two steps. The first

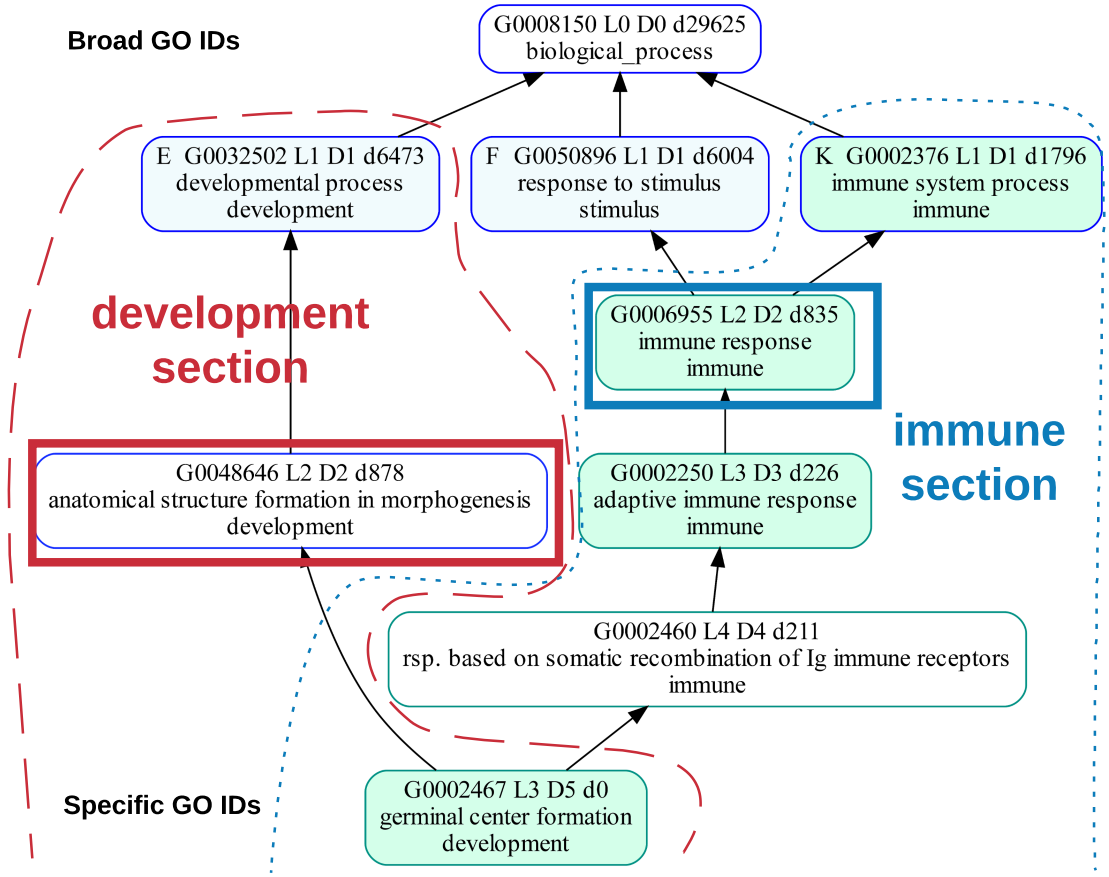

Figure 5.6: **The GO term *germinal center formation* (bottom green term) can be in the *development* section (dashed red line), the *immune* section (blue dotted line).** Section membership for each GO ID is also specified by the text at the bottom of each GO term box. By default, there are five potential GO headers for *germinal center formation* (GO terms with a blue border). The *germinal center formation* is initially in the *development* section because the GO header *anatomical structure formation in morphogenesis* (boxed in red) has a dcnt of 878 while the next lowest GO header is *immune system response* with a dcnt of 1796. But if *immune response* (boxed in blue) is added as a new GO header to the *immune* section, it pulls *germinal center formation* into the *immune* section. The text at the top of each GO term box is described as follows. The total number of GO terms below the GO term box is indicated by the number next to the 'd'. Level, the minimum path from the top root term is indicated by the number next to the L. Depth, the maximum path from the top root term is indicated by the number next to the D. For example "L3 D5" on *germinal center formation* indicates that the minimum path is 3 (through the through the development section) and the maximum path is 5 (through the immune section).

grouping step uses broad GO terms as GO headers, where the contents of the group under a GO header are the header's descendant GO terms that are also GOEA results. The second grouping step uses researcher-created section titles like *immune* and *neurological* as section headers, where the contents of each section are GO headers from the first step.

Two-step grouping method

The default list of GO headers used in the first step of the GOATOOLS grouping is the list of species-agnostic generic GO slims from the Gene Ontology Consortium (`http://www.geneontology.org/ontology/subsets/goslim_generic.obo`). As of April 2018, there are over 200 GO IDs in the GOC's generic GO slim list out of the over 47,000 GO IDs in the full *go-basic.obo*. To take full advantage of GOATOOLS grouping, researchers will likely want to add additional broader GO terms as GO headers.

Having user-created section titles used in the second step is necessary because not all terms that researchers may want to view together lie in one GO branch. For example, *immune system process* and *T cell apoptotic process* are in parallel branches that only intersect at the topmost term, *biological_process*. Without a sections list, the GO terms associated with both of these GO headers could end up in separate areas of the results list. The sections list ensures that all GO IDs under both GO headers of interest appear in one area of the grouped GOEA results list. Researchers may also use the sections list to group GO IDs from under different top-level branches, like *biological_process* and *molecular_function*.

GO terms frequently have multiple parents. To print a GO term just once, regardless of how many parents it has, GOATOOLS chooses the "most specific" GO header parent under which to place the GO if it has multiple parents. The user may override this default by adding new GO headers to the sections list. The "most specific" GO header can be determined using either information content (tinfo) or descendants count (dcnt). A user-created function may also be used to determine the most specific header GO term.

Assigning sections

Section names are user-specified descriptive text, not GO terms, created using research questions or based on interesting GO terms found enriched in the GOEAs.

For example, the Gjoneska research questions involved immune and neurological functions. So I create two sections, *immune* and *neuro*. Because the GOEA results for the Gjoneska data showed a number of GO terms related to virus and bacteria, I added a *virus/bacteria* section.

I created one sections file by grouping the more than 800 total GO terms found significant from the combined findings of GOATOOLS, DAVID6.7, DAVID6.8, and GOstats GOEA runs using all Gjoneska study gene sets. After this single sections file is generated, it is then reused twenty times: once for each of the five Gjoneska clusters showing significant GO terms for all four tools.

Assigning sections details

To begin creating the sections file for the entire project, I run the *wr_sections* script on the list of the 800+ project GO terms, stored in *goids_all.txt*. An initial sections file *sections_in.txt* is written because there was none to be read.

```
$ wr_sections.py goids_all.txt
hdr GOs(0 in 0 sections,  61 unused) WROTE: sections_in.txt
hdr GOs(0 in 0 sections,  61 unused) WROTE: sections.txt
usr GOs(0 in 0 sections, 840 ungrpd) WROTE: grouped_gos.txt
   840 user GO IDs
```

Three files are written for this initial run:

1. ***sections_in.txt*** This file is read if it exists and written if it does not exist. If it is written, all GO headers which currently represent the GO IDs listed in *goids_all.txt* are written into the ungrouped area listed under *Misc.* In my studies, 61 GO headers represent the 800+ GO terms.

2. ***sections.txt*** This file is always written. It is *sections_in.txt*, but with the ungrouped section (*Misc.*) recalculated and all GO IDs annotated with depth, dcnt, name, etc.

3. ***grouped_gos.txt*** This file is always written. It contains the current grouping of the user GO IDs as guided by *sections_in.txt*. On the first run, zero user GO terms are grouped and 840 GO terms are ungrouped.

The initial *sections_in.txt* contains a GO ID related to immune, *immune system process*, in the ungrouped area:

```
# SECTION: Misc.
# GO ID      NS hdrusr # user  dcnt level depth GO name
----------  --- -----  ------ ----- ----- ----- --------------------
```

```
GO:0008150 # BP    **  1 uGOs 29625   L00   D00 biological_process
GO:0009987 # BP    **  7 uGOs 18703   L01   D01 cellular process
...
GO:0002376 # BP    ** 79 uGOs  1796   L01   D01 immune system process
...
```

To begin to group the immune GO IDs, add a new section *immune* in the *sections_ini.txt* and move the *immune system process* GO term into the new section:

```
# SECTION: immune
GO:0002376 # BP    ** 79 uGOs  1796   L01   D01 immune system process

# SECTION: Misc.
# GO ID      NS hdrusr # user  dcnt level depth GO name
----------  --- -----  ------ ----- ----- ----- --------------------
GO:0008150 # BP    **  1 uGOs 29625   L00   D00 biological_process
GO:0009987 # BP    **  7 uGOs 18703   L01   D01 cellular process
...
```

Moving this one GO header, *immune system process* into the *immune* section in *sections_in.txt* causes the *wr_sections* script to move 79 of the 800+ into the immune section:

```
$ wr_sections.py goids_all.txt
hdr GOs( 1 in 1 sections, N/A unused) READ:  sections_in.txt
hdr GOs( 1 in 1 sections,  60 unused) WROTE: sections.txt
usr GOs(79 in 1 sections, 761 ungrpd) WROTE: grouped_gos.txt
```

GO header and sections decisions can also be made based upon the current GO grouping in *grouped_gos.txt*. For example, the *grouped_gos.txt* file shows many GO IDs related to *interleukin* and *interferon* are ungrouped. These GO terms fall under the broad GO:0001816, *cytokine production*. Adding GO:0001816 under the *immune* section adds it as a new GO header.

```
# SECTION: immune
GO:0002376 # BP    ** 79 uGOs  1796   L01   D01 immune system process
GO:0001816 # cytokine production
```

```
# SECTION: Misc.
# GO ID      NS hdrusr # user  dcnt level depth GO name
----------  --- -----  ------ ----- ----- ----- --------------------
GO:0008150 # BP    **  1 uGOs 29625   L00   D00 biological_process
GO:0009987 # BP    **  7 uGOs 18703   L01   D01 cellular process
...
```

Adding *cytokine production* results in the placement of 19 additional user GO IDs into the *immune* section for a total of 98 grouped GO IDs:

```
$ wr_sections.py goids_all.txt
hdr GOs( 2 in  1 sections, N/A unused) READ:  sections_in.txt
hdr GOs( 2 in  1 sections,  60 unused) WROTE: sections.txt
usr GOs(98 in  1 sections, 742 ungrpd) WROTE: grouped_gos.txt
```

To discover that *cytokine production* may be the appropriate GO header to represent *interferon* and *interleukin*, the GO DAG can be queried either by creating a plot or a report. To create a plot containing user relevant GO IDs:

```
# 1) CREATE A PLOT CONTAINING interferon & interleukin GO IDs
# Create a list GO IDs that match 'interleukin' or 'interferon'
$ grep inter grouped_gos.txt > inter.txt
# Plot the list of GO ID
$ go_plot.py -i inter.txt -o inter.png --sections=sections.txt
```

To create a report of the GO Terms up the hierarchy starting from GO:0032611, *interleukin-1 beta production*, up to the root term GO:008159, *biological_process*:

```
# 2) REPORT GO:0032611 "interleukin-1 beta production" to root
$ wr_hier.py --up GO:0032611
  - GO:0008150   29625 D00 biological_process
  -- GO:0032501   7544 D01 multicellular organismal process
  --- GO:0001816   110 D02 cytokine production
  ---- GO:0032612    2 D03 interleukin-1 production
> ----- GO:0032611   0 D04 interleukin-1 beta production
```

From the printed report, *cytokine production* in the middle of the hierarchy list is a good term to represent bottom term, *interleukin-1 beta production*,

because *interleukin-1 production* just below *cytokine production* is too specific and *multicellular organismal process* just above is too broad.

In my example, moving just two GO headers, *immune system process* and *cytokine production*, resulted in the placement of 98 user GO IDs into the *immune* section. Most GO header movements will result in smaller numbers of user GO IDs grouped, but the method remains the same.

Researcher-guided grouping method

The human element of a researcher's subjective input by grouping and describing GO terms can lead to visualizing information in a unique way, which can lead to unexpected insights.

One reason for the need for the researcher's insight is that a GO term can be accurately described using multiple, and potentially subjective interpretations. For example, *germinal center formation* may be correctly described as being both a *developmental process* and also related to the *adaptive immune response* (Figure 5.6).

Germinal centers are a *developmental process* because they are transient structures that develop in the sites of secondary lymphatic organs, such as lymph nodes, during an immune response [128]. Germinal centers are an *adaptive immune response* because inside germinal centers, B cells proliferate expeditiously with the immunoglobulin variable region of the new B cells diversified by somatic hypermutation, resulting in the production of new generations of high affinity memory and plasma B cells [42].

By default, *germinal center formation* (green GO term at the bottom of Figure 5.6), is grouped in the *development* (under red dashed line) section rather than

the *immune* section (under blue dotted line) due to it favoring the goslim GO header, *anatomical structure formation in morphogenesis* (red box). Section names for each GO are also indicated by the text at the bottom of each GO box.

The GO header, *anatomical structure formation in morphogenesis* (red box), is chosen from five possible GO headers (GO terms with a blue border) to represent *germinal center formation* because it has the smallest dcnt value (878) compared to 1796, 6004, 6473, and 29625. To move *germinal center formation* from *development* to *immune*, add a new GO header, *immune response*, (blue box) to the *immune* section. The GO term, *germinal center development*, will then be moved to the *immune* section because *immune response* has a dcnt of 835 which is less than 878.

Knowing that the research question concerns the role of the immune system in a particular condition and seeing numerous GOEA results in "immune", a researcher may wish to guide a GO grouping of GOEA results such that a succinct summary clearly highlights the immune findings and the genes associated with those immune findings. Alternately, the researcher may prefer to highlight only the developmental aspect of germinal centers or both high-level descriptions, *developmental process* and *immune response*, at the cost of duplicating the GO term which makes the results list longer. Grouping is used to organize an already long list of GO terms to make the results easier to interpret, so making the list even longer may not be desired.

5.3.8 GOATOOLS grouping vs. ReviGO visualization

GOATOOLS grouping can be preferable to tools such as ReviGO (Reduce and Visualize Gene Ontology) [201] if the researcher wants to retain the full list of

GO IDs returned from a GOEA, but organize the list so GO IDs are stored under large user-defined sections.

If a graphical visualization of the overall properties of all user GO IDs is desired, ReviGO is an excellent tool that can help visualize GO groupings using scatter plots, interactive graphs, and tag clouds. GOATOOLS grouping is list-based only. GOATOOLS GO plots are a tool for GO header placement decisions in the sections file and not considered a final output for an entire list of GO IDs.

ReviGO is desirable when the researcher wishes to reduce a list of GO terms using ReviGO's redundancy reduction. GOATOOLS grouping philosophy is to retain the full list of GO IDs.

GOATOOLS grouping also allows the user to move groups of GO IDs from one section to another. This is necessary because GO terms can be correctly represented under more than one section. A researcher may wish to guide the specific section for the placement of the GO IDs using the research hypotheses and the GOEA results.

GOATOOLS grouping is preferable if the researcher wishes to retain the full list of statistically significant GO IDs, have control over choosing from multiple equally valid grouping decisions, and prefers to see the GO IDs in a list rather than a figure.

5.3.9 Example usage of the Python API

An example of code which groups GO IDs into user-created sections is as follows, with many more code examples available as Jupyter notebooks on Github:

```
import collections as cx
from goatools.test_data.gjoneska_goea_consistent_increase import goea_results
from goatools.test_data.sections_gjoneska import SECTIONS
from goatools.grouper.grouper import wr_xlsx_gos

xlsx1 = "goids_consistent_increase.xlsx"
xlsx2 = "goids_consistent_increase_dcnt.xlsx"
# NtGoeaResults = cx.namedtuple("NtGoeaResults", "GO p_fdr_bh name ...
# goea_results = [
#     NtGoeaResults(GO='GO:0035458', p_fdr_bh=4.21e-07, name='cellular response to ...
#     NtGoeaResults(GO='GO:0002376', p_fdr_bh=4.32e-07, name='immune system process',
#     NtGoeaResults(GO='GO:0006954', p_fdr_bh=4.74e-07, name='inflammatory response',
#     ...
goids = [nt.GO for nt in goea_results if nt.p_fdr_bh < 0.05 and nt.enrichment == 'e']

# SECTIONS = [ # 18 sections
#     ("immune", [ # 15 GO-headers
#         "GO:0002376", # immune system process
#         "GO:0002682", # regulation of immune system process
#         "GO:0030155", # regulation of cell adhesion
#         ...
#     ]),
#     ("viral/bacteria", [ # 4 GO-headers
#         "GO:0016032", # viral process
#         "GO:0050792", # regulation of viral process
#         "GO:0098542", # defense response to other organism
#         ...
#     ]),
#     ...

# GROUPING OPTION #1:
# The most specific GO header is determined using information content
# calculated using the annotations.
wr_xlsx_gos(xlsx1, goids, sections=SECTIONS, gaf='gene_assocation.mgi')

# GROUPING OPTION #2:
# The most specific GO header is determined using descendants count.
wr_xlsx_gos(xlsx2, goids, sections=SECTIONS)
```

5.3.10 Case study: Gjoneska dataset

I used the Gjoneska gene expression data to compare the GOEA results among four different tools. The first tool was the older DAVID version 6.7 released in 2010 [73] and is referred to as "DAVID6.7". The second was the most recent version of DAVID version 6.8, a major update completed in October 2016 and is referred to as "DAVID6.8". The third set of GOEA results was generated by running GOstats, an extremely popular tool for running gene ontology

analyses using the statistical language, R. The fourth set of GOEA results was generated by running GOATOOLS v0.8.2 [21].

Versions of ontologies, annotations and tools

I used the following versions of ontologies, annotations, and tools for the four utilities analyzed in this paper. First, my DAVID6.7 analyses use the DAVID Knowledgebase released Sep 2009 with version 6.7 of the DAVID software released Jan 2010. Second, my DAVID6.8 analyses use the DAVID Knowledgebase released May 2016 with version 6.8 of the DAVID software released Oct 2016. Third, my GOstats analyses use GO.db from NCBI gene Sep 21, 2016 and org.Mm.eg.db version 3.4.0 released Oct 2, 2016. The GOstats software used is in Bioconductor version 3.32 (Oct 31, 2016). Fourth, my GOATOOLS analyses use the ontologies in go-basic.obo released Apr 21, 2018 and annotations from gene_association.mgi released Apr 2, 2018. Finally, GOATOOLS grouping used GO slims from goslim_generic.obo downloaded Apr 22, 2018. All GOATOOLS analyses were run using GOATOOLS version 0.8.2 released Feb 22 2018.

To generate the DAVID6.7 GOEAs, I used the DAVID annotation set, GOTERM_BP_ALL, because it was used to generate the GOEA data found in Gjoneska's Supplemental Table 2. Also, GOTERM_BP_ALL was the set of annotations available in DAVID which produced results closest to the GOATOOLS GOEA results.

To generate the DAVID6.8 GOEAs, I used the newly available GOTERM_BP_DIRECT annotations because it is the original unmodified annotations, which is what I used for all GOATOOLS analyses in this paper.

The GOTERM_BP_ALL annotations augment the original annotations by propagating parent GO terms up the hierarchy.

For my cross-tool comparisons, I had to use two different sets of DAVID annotations because GOTERM_BP_DIRECT was not available in DAVID6.7. So I added a comparison of DAVID6.7 and DAVID6.8 using GOTERM_BP_ALL for both to examine the effect of using old versus new annotations.

Gjoneska data set

Gjoneska's gene expression data was used to investigated immunity in Alzheimer's disease using mice that can be induced to display Alzheimer-like extreme neuronal loss and increased beta-amyloid peptide production and tau pathology [73].

Gjoneska measured the gene expression of cells in the hippocampus at early (2 weeks after induction) and late stages (6 weeks after induction) after inducing the Alzheimer model. The Gjoneska gene expression results are organized into upregulated and downregulated genes at three time-points each. The first of the three time-points is *Transient* indicating the gene expression was only seen in the early stage (2 weeks). The second time point, *Consistent*, indicates the gene expression was seen in both early and late stages (both 2 and 6 weeks). The third time point, *Late*, indicates the gene expression was seen in only the late stage (6 weeks).

Gjoneska found an upregulation of immune genes and a downregulation of synaptic plasticity genes. I compare the upregulated immune results found in the Gjoneska paper using all four tools. The genes I examined are the upregulated genes in the three clusters; *Transient Increase* (TI), *Consistent*

Increase (CI), and *Late Increase* (LI). Immune and viral or bacterial functions of statistical significance were the focus for my studies across the four tools. The population and study gene sets used in my GOEAs are from Gjoneska's supplementary table one, *Gene expression differences in the CK-p25 mouse*.

5.4 Results and Discussion

I compared GOEA results from GOATOOLS and the other tools through both stochastic simulations as well as real-world case study from Gjoneska et al. Overall, I show that GOATOOLS provides GO terms by median descendant count are twenty times more specific than the broad GO terms from DAVID6.7, two times more specific that GOstats, and similar in specificity to DAVID6.8 GO terms.

5.4.1 Stochastic simulation study

GOATOOLS GOEA performance was tested by running 100,000 stochastic gene ontology enrichment analyses (GOEAs) simulations (Figure 5.4 and Supplemental Figures S1-S3).

Each simulation tested the correctly identified enrichment in a stochastically generated study gene list whose size ranged from four to 124 genes against a population of more than 20,000 mouse protein-coding genes. The study gene lists contained two types of randomly chosen genes: target genes and background genes [48]. Uncorrected P-values were generated using Fishers exact test. Corrected P-values were generated using Benjamini/Hochberg multiple test correction.

The target gene pool contains 124 genes associated with the humoral response (HR) biological process. The background gene pool contains the entire list of protein-coding genes excluding HR genes. One study set of genes contains one of the following percentages of background genes, also known as *Null* or *True Null* genes: 0%, 25%, 50%, 75%, and 100%. A study set of 100% null genes contains genes chosen only from the background set (Figure 5.4, row 1). A study set of 0% null genes contains only randomly chosen HR genes (Figure 5.4, row 5). A study set of 16 genes containing 25% null genes contains 4 randomly chosen background genes and 12 randomly chosen HR genes (Figure 5.4, row 4). The target genes function as true positives in the GOEA while the background genes are counted as false positives.

5.4.2 Simulation study results

The first simulations contained unacceptably high FDRs for larger study gene groups (Supplemental Figure S1). Upon investigation of the failing FDRs, there were two characteristics of GO IDs that were associated with the false positive study genes. First, the GO IDs related to the false positives are associated with thousands of genes. This is contrasted to the statistics for the overall mouse protein-coding associations: median=3 genes/GO; mean=16 genes/GO, and stddev=128. Second, the GO IDs are much more likely to be under-represented, rather than enriched. An under-represented term is one in which far fewer genes appeared significant in the study set than in the general population.

Upon running the simulations viewing only *enriched* gene lists, the simulations solidly passed resulting in FDRs that were nearly zero (Figure 5.4). Only 30 GO IDs out of over 17,000 GO IDs associated with mouse

protein-coding genes are associated with over 1,000 genes. Upon running the simulations using an association with ~30 GO IDs pruned out of the association, the simulations also passed with FDR values close to zero.

Performing stress tests by randomly shuffling the associations for True-Null genes prior to simulation, the "view-enriched-gene" simulations either passed or were very close to passing (Supplemental Figure S2) and all "30-GOs-Purged" simulations passed (Supplemental Figure S3).

The results of the GOATOOLS GOEA simulations show excellent FDR and specificity levels (Figure 5.4A). The sensitivity varied with studies having 64+ genes performing well and study sizes of 4 genes performing poorly (Figure 5.4A, panels B2 to B5), where truly enriched genes were not identified. Study sets containing 16 gene study sets performed well if 75%+ of the 16 study genes were truly enriched (Figure 5.4A, panels B4 and B5). These simulation results are true only when viewing genes associated with statistically significant GO IDs that are enriched, not under-represented. Adding genes associated with under-represented GO terms resulted in an unacceptably high ratio (> 0.05) of genes seen as associated with significant functions (Supplemental Figure 1). The GOEA sensitivity is greatly improved, especially for small (4-20 genes) gene groups, if the option *propagate_counts* is set to "True," which updates the annotations such that a gene's associated GO terms now include all parent GO terms (Figure 5.4B, panels B4 and B5). To compare results among the four tools, *propagate_counts* is set to the more conservative value, "False," in GOATOOLS GOEAs which causes the annotations to be used in their original form with no modifications.

To recreate all five of my stochastic GOEA simulation plots (for a total of 100,000 total stochastic simulations) featured in the GOATOOLS manuscript and supplemental data, clone the repository, `https://github.com/dvklopfenstein/goatools_simulation`, and run this make target from the command-line:

```
$ make run_ms
```

Generating the five simulation plots in the GOATOOLS manuscript and supplemental data takes about 38 hours on a laptop PC running an Intel(R) Core(TM) i7-6500U and 16GB of RAM.

5.4.3 Counts of genes associated with statistically significant GO terms

GOATOOLS and DAVID6.7, and DAVID6.8 total gene counts for the Gjoneska study sets are much more similar (2,335, 2,443, and 1,988 respectively) than gene counts for GOstats (3,652 total genes) (Figure 5.7).

Looking closer, if I remove only *cellular process* (GO:0009987), an extremely broad depth-01 term with more than 18,000 GO term descendants out of a total of more than 29,000 in the entire *biological process* branch, the total genes associated with significant GO IDs reduce from 3,652 to 3,310 (342 genes removed) for GOstats (Supplemental Figure 4). The genes that are removed are associated with *cellular process* and with no other GO IDs in the GOstats GOEA results. Removing *cellular process* has no effect on the GOATOOLS or DAVID6.8 results which do not show significance for *cellular process* although there are numerous more specific GO terms under *cellular process* that are statistically significant.

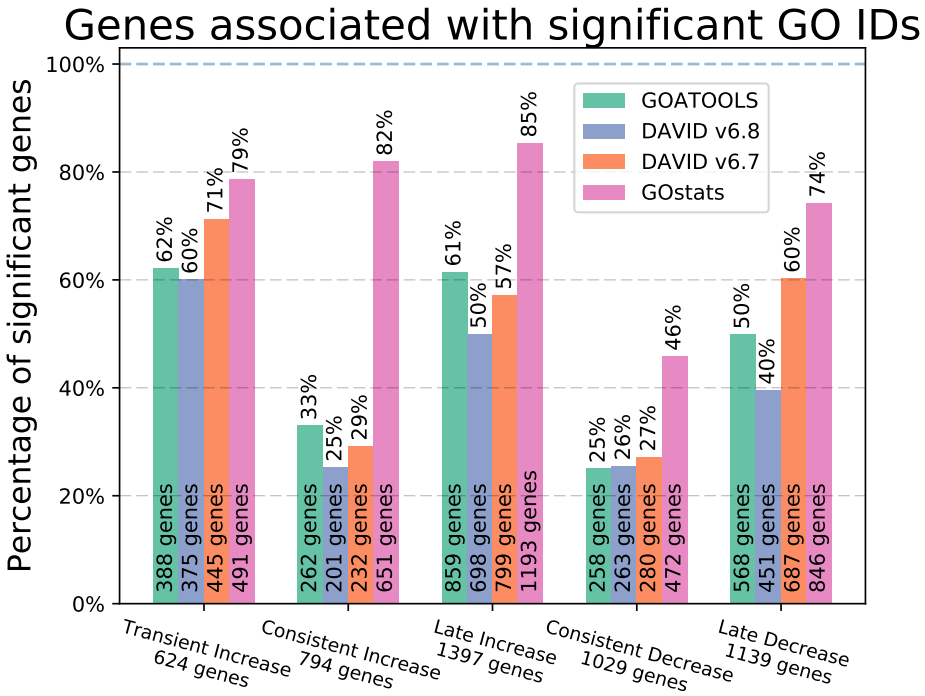

Figure 5.7: **Percentages of genes associated with statistically significant GO terms for all tools and all Gjoneska clusters.** The GOEA analyses from four different tools found significant GO IDs for five of the six Gjoneska gene clusters using the Gjoneska population background of 13,838 genes. The x-axis shows the five Gjoneska clusters and the total count of genes found to be up or down regulated in the Gjoneska experiments. The number of genes in each cluster that are found to be associated with significant GO IDs for each tool is printed at the bottom of each tool bar. The color of each bar represents a GOEA tool as specified in the legend. The height of each bar is the percentage of genes in each cluster that are found to be associated with significant GO IDs. GOATOOLS is most similar to the DAVID tools. GOstats found between 74% and 84% of the genes significant for four clusters, which will be reduced if the statistically significant but extremely broad term, *cellular process*, is removed.

In practice, I might consider the large list of genes directly associated with *cellular process* from GOstats rarely useful. There are 57 GO terms in DAVID6.7 and 57 GO terms in GOstats which are both broad (meaning the descendants count is over 200) and have associations of more than 100 study genes. There are no such GO terms in GOATOOLS and only 3 in DAVID6.8. Therefore, it may be desirable to not include some of these broad terms in a GOEA summary.

5.4.4 Broad vs. specific GO terms by grouping

The four tools together found a total of 833 GO terms statistically significant. I first grouped the GO terms into sections using the popular annotation-associated value, *information content* (*tinfo*), and then created a second grouping using the species and tool agnostic value, *descendants count* (*dcnt*).

The two grouping methods showed strong concordance with 810 GO IDs (97%) agreeing on section placement. There was disagreement in section placement for 23 GO IDs (2.76%). One example of disagreement was that the GO term, *transmission of nerve impulse*, was placed in the *neurological* section using *dcnt* and *signaling* using *tinfo*. A second example of disagreement was that the GO term, *trophoblast giant cell differentiation*, was placed in the *reproduction* section using *tinfo* and the *Misc.* (uncategorized) section using *dcnt*. The researcher may override any of these section placements by adding more specific GO headers to place the GO IDs of interest into a section which better informs the research question.

I chose to use *dcnt* rather than *tinfo* to compare GO results because *tinfo* values are determined using a set of annotations. But the four annotation sets differed among the four tools as revealed by the GO terms having different sets of genes in their associations. To avoid choosing annotations used by a single tool to evaluate all tools, I used *dcnt* to group the GO terms. In general *tinfo* may preferable to use in grouping decisions because it is determined by the annotations. But *dcnt* can be used if comparing sets of GO terms whose annotations differ, like when comparing sets of GO terms between different species or different tools.

In general, GOATOOLS GOEA GO ID results were consistently much closer to leaf-level, as measured by descendant count, than the DAVID6.7 results and the GOstats results (Figure 5.8A). GOATOOLS and DAVID6.8 both discovered GO IDs with the lowest and most similar specificity. GOstats median GO ID broadness was twice that of GOATOOLS and DAVID6.8. DAVID6.7 discovered GO IDs 10x as broad as GOATOOLS when using mean *dcnt*.

The GOstats results were much lower than the DAVID6.7 results with a median descendant count of 35 (mean=491, SD=2,198) for GOstats compared to a median of 211 (mean=775, SD=1,678) for DAVID6.7. GOATOOLS (median=19, mean=199, SD=781) and DAVID6.8 descendants counts (median=20, mean=197, SD=711) distributions were most similar among all tools examined. I saw a trend where GO terms closer to the bottom of the DAG, terms considered to be more specific, are associated with fewer genes (Figure 5.8B). All four tools show a positive correlation between descendants count and the number of genes associated with the GO term. GOATOOLS and

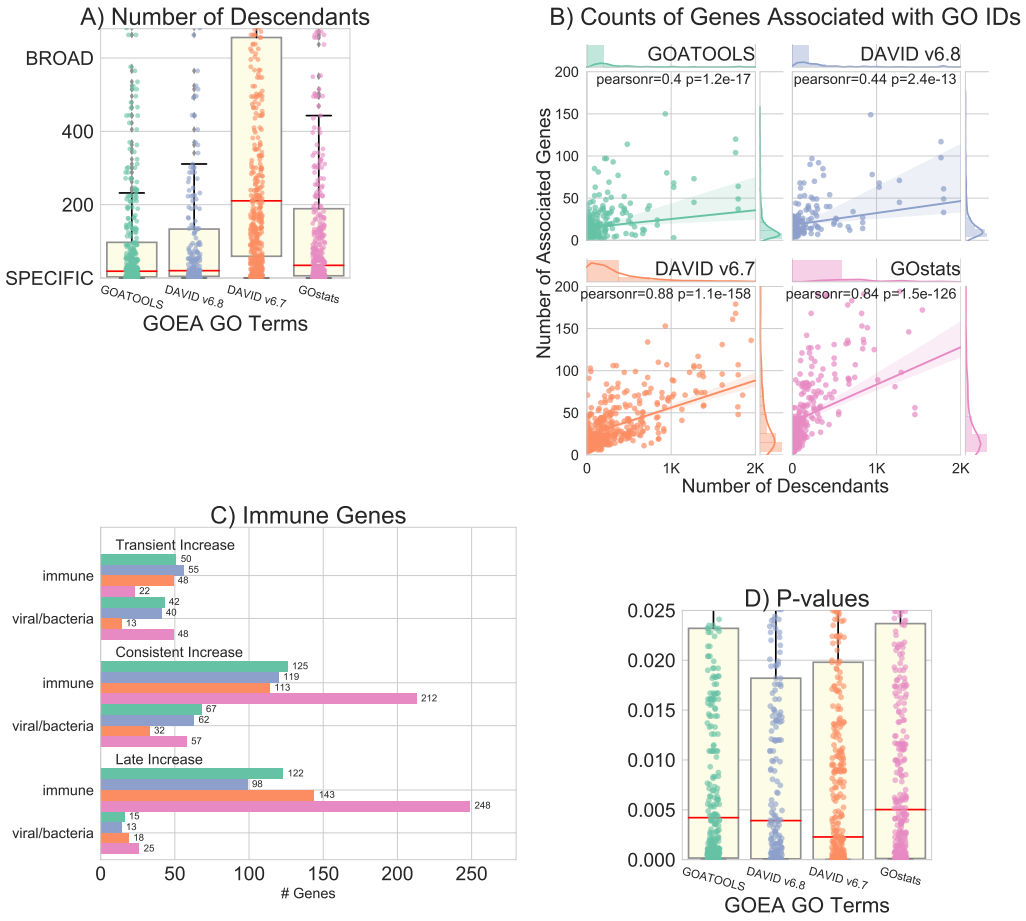

Figure 5.8: **Comparison between enriched terms identified by GOATOOLS, DAVID and GOstats.** All panels use the the same color coding as specified in the legend in Figure 5.7. (A) Number of descendants for the significant terms reported in GOATOOLS, DAVID6.7, DAVID6.8, and GOstats. (B) Broader GO terms are associated with more genes while specific GO terms are associated with fewer genes. (C) Clusters and counts of genes significant for terms related to immunity. GOATOOLS and DAVID6.8 are most similar in the types and numbers of genes discovered. (D) Comparison of P-values for all of the GO terms found in total in all four tools. The mean P-values were similar for GOATOOLS, DAVID6.8, and GOstats. DAVID6.7 had P-values multitudes lower than all the other tools.

the most recent DAVID generally discovered very specific GO IDs associated with fewer genes. DAVID6.7 and GOstats found broader GO IDs that were associated with large numbers of genes.

5.4.5 Example functional groups: immunity and viral/bacteria

Genes associated with statistically significant immune GO terms were found in all three upregulated clusters by all four tools (Figure 5.8C). GOATOOLS discovered genes significant in immune and viral/bacterial categories for all Gjoneska clusters contrasted to the DAVID6.7 which found no viral/bacteria genes for any cluster. GOstats often found more genes, but they were often associated with broad GO IDs.

To view the results of the GOEAs, I chose to split GO terms related to virus or other parasites into their own *viral/bacteria* section. Genes associated with *viral/bacteria* were found in all three clusters by all tools. GOATOOLS GOEAs found 42, 67, and 15 study genes with statistically significant GOs in the *viral/bacteria* section in all three clusters: *Transient*, *Consistent*, and *Late Increase*. GOstats found more genes than GOATOOLS (48, 57, and 25).

The GOATOOLS GOEAs found 50, 125, and 122 study genes associated with statistically significant GOs in the *immune* section in three clusters: *Transient*, *Consistent*, and *Late Increase*. DAVID6.8 found slightly fewer genes (55, 119, 98) than GOATOOLS. GOstats found generally more genes than GOATOOLS in the clusters (22, 212, 248).

GOATOOLS and DAVID6.8 reported similar number of associated genes no matter the level of the GO. GOATOOLS and DAVID6.7 reported different number of associated genes no matter the level of the GO. Curiously,

lymphocyte aggregation, with a very low dcnt of 5 found significant by GOstats, but not by GOATOOLS was associated with 46 genes (Supplemental Table 3). Although it failed to reject the null hypothesis by GOATOOLS, *lymphocyte aggregation* was only associated with one gene in the association from MGI.

5.4.6 Differences among tools

As an example of evaluating the differences between the results from the four tools, I describe the statistically significant GO terms in the *immune* section for the *Consistent Increase* cluster comparing GOATOOLS vs DAVID6.7 (Supplemental Table 1), DAVID6.8 (Supplemental Table 2), and GOstats (Supplemental Table 3). These tables are each sorted by descendant counts such that broader terms are listed before specific terms.

The DAVID6.7 terms tend to be concentrated at the top among the broader terms while missing specific GO terms at the bottom of the table that were found by GOATOOLS (Supplemental Table 1). GOATOOLS found 6 broader terms also found by DAVID6.7. But most terms found by GOATOOLS are extremely specific having a dcnt less than or equal to 11. For example, *positive regulation of interleukin-1 beta secretion*, with a depth of 11 in the third row from the bottom of the table is significant in the GOATOOLS GOEA and is associated with nine genes in the study. The statistically significant GO terms in the *immune* section are associated with a total of 125 genes as found by the GOATOOLS GOEA and 113 genes as found by the DAVID6.7 GOEA. The asterisk in most of the GOATOOLS P-value column indicates that where DAVID6.7 found a broader term significant, GOATOOLS found a more specific term in that term's descendants significant.

DAVID6.8 performs much more similarly to GOATOOLS (Supplemental Table 2). When both GOATOOLS and DAVID6.8 find the same GO term, the number of associated genes is similar for the two tools indicating the associations used by the two tools are similar. GOATOOLS finds more GO terms significant than DAVID6.8. GOATOOLS finds GO terms that are more specific than found by DAVID6.8.

In the comparison between GOATOOLS and GOstats (Supplemental Table 3), the more specific bottom half of the table has similar GO term findings between the two tools. Additionally, in the bottom half of the table, when both GOATOOLS and GOstats find the same GO term, the number of associated genes is similar. The top half of the table showing the broader GO terms is where I see the larger differences between GOATOOLS and GOstats. The largest difference seen in the top half of the table is when GO terms are found by both tools, GOstats reports many more study genes associated with the GO term than reported by GOATOOLS.

5.4.7 GO term overlaps among tools

The total counts of significant GO terms found by GOATOOLS, DAVID6.8, DAVID6.7, and GOstats is 383, 230, 390, and 428, respectively. GOATOOLS found the same GO IDs as DAVID6.8, DAVID6.7, and GOstats in the quantities of 227, 110, and 206. GOATOOLS and DAVID6.8 had the most concordance. GOATOOLS found hundreds of more specific GO terms than in DAVID6.7. But in DAVID6.8, the specificities of the GO terms were well-matched with those of GOATOOLS.

Examples of terms that are close to leaf-level found by GOATOOLS, but not found by DAVID6.8 or GOstats in the *Late Increase* cluster include *toll-like receptor signaling pathway*, *natural killer cell differentiation*, "*complement activation, classical pathway*," and *neutrophil chemotaxis*. Both GOATOOLS and GOstats found *GO:0045576 mast cell activation* significant while DAVID6.8 did not.

Sometimes, one tool would find significance in a broader term that was not found by GOATOOLS. However, that broader term was actually covered by GOATOOLS by finding more specific children under the "missing" broader term. For example, "*antigen processing and presentation*" is found in GOstats and DAVID6.7 but not in GOATOOLS. But the more specific GO term under it, "*antigen processing and presentation of endogenous peptide antigen via MHC class I via ER pathway, TAP-dependent*," was found statistically significant only in GOATOOLS. The counts of broader terms found by other tools that were actually covered by GOATOOLS finding more specific children terms are 166 for GOstats, 10 for DAVID6.8, and 291 for DAVID6.7.

At times, broad GO terms with low information content (i.e. associated with large quantities of gene products) found in GOEAs may not meaningfully map to the more specific GO terms. For example, GOstats found *cellular process* significant (purple, on left) in Figure 5.9. The leaf-level GO term *cellular response to interferon-beta* (green) has three potential GO headers (top three purple GO terms) that are extremely broad. Even *response to stimulus* is broad because its descendants are as diverse as *response to gravity*, *startle response*, and *immune response*. The user may wish to add a new GO header which better represents *cellular response to interferon-beta*, like *response to cytokine* (circled in blue). Colors in the GO term boxes indicate if one (purple) or more (green)

tools found a GO term significant. GOATOOLS, DAVID6.8, and GOstats found *cellular response to interferon-beta* (green) significant. GOstats found the broad terms, *biological_process*, *cellular process*, and *response to stimulus* significant (purple). The blue GO term borders indicate that a GO term is also a GO header. Of the GO terms pictured here, only *biological_process* is found in the GO slims as of April 2018. The depth-01 GO terms are default header GO terms because GOATOOLS grouping adds all depth-01 GO terms to the list of default headers.

The purple terms are terms found significant in GOstats but not found significant in either DAVID6.8 or GOATOOLS. A specific GO term that could be represented by *cellular process* is the leaf-level term *cellular response to interferon-beta* (green), which is found significant in GOATOOLS, GOstats, and DAVID6.8. The purple GO header terms are so broad that I cannot be sure that they meaningfully cover the specific GO term, *cellular response to interferon-beta* (green, bottom). Even *response to stimulus* (purple, top right) is an extremely broad umbrella term encompassing terms as diverse as *eye blink reflex* and *innate immune response* and is not a meaningful proxy to represent *cellular response to interferon-beta*.

5.4.8 Summary

GOATOOLS results were most similar to DAVID6.8's results when using DAVID's new GOTERM_BP_DIRECT GO set in terms of numbers of genes found, the P-value values, and the similarity of the GO term specificity. GOstats and DAVID6.7 found more broad terms, but that is likely because they

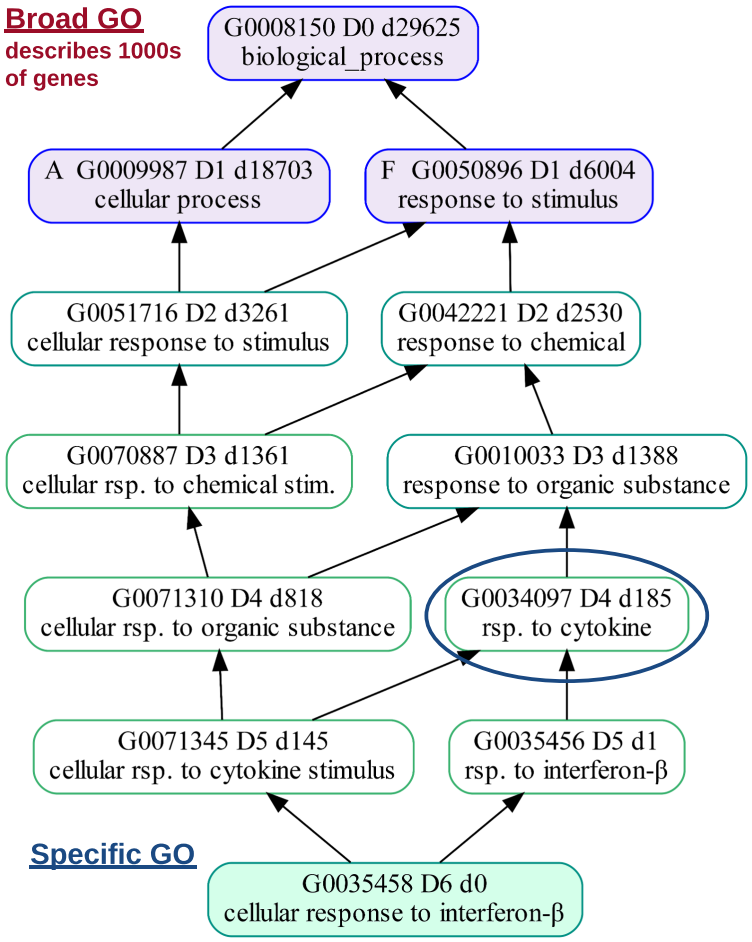

Figure 5.9: **Leaf-level GO terms may initially have broad GO headers that do not convey enough information about a leaf-level term.** The leaf-leve GO term, *cellular response to interferon-beta* (bottom green term), has three potential GO headers (top purple terms) that are extremely broad.

employ some form of propagate counts to augment the original association. DAVID6.7 misses many specific terms.

Using the latest ontologies and annotations

The main difference between the GOATOOLS GOEA results and the DAVID6.8 results was that GOATOOLS found specific GO terms not found by DAVID6.8. This could be a consequence of being able to use the very latest ontologies and annotations in GOATOOLS, a crucial factor that influences all GO term enrichment results and is described in a recent article by Wadi et al., in Nature Methods [210]. Wadi reports that using old annotation datasets and old ontology versions has influenced the results of thousands of recent studies by markedly underestimating the functional significance of their gene lists, negatively affecting follow-up studies.

Both the ontologies and the annotations change daily, with the number of human biological processes doubling from 6,509 in 2009 to 14,735 in 2016 [210]. In the ontologies, the GO vocabulary is increasingly expanded, resulting with GO terms having longer paths to roots and terms having more parents. Some GO terms are rendered obsolete and are pruned as biological knowledge expands. The number and quality of annotations per gene constantly increases with high-confidence experimental annotations becoming more frequent, and more genes annotated. Poor quality annotations are pruned due to constant quality control efforts. Annotations can vary among tools.

That GOATOOLS and DAVID6.8 performed with greater similarity than DAVID6.7 could be due to the fairly recent update to DAVID.

One of the greatest benefits of using GOATOOLS is that the researcher has full control of the versions of the GO and the annotations that he or she uses. The full GOEA project can be archived including the ontology, annotation, study and population gene product sets, and code used to generate the GOEA results since all text files and code and accessible to the researcher.

Effects of old and new annotations in DAVID

To get a sense of how the dates of the GO DAG and the annotations might affect GOEA results, I compared DAVID6.7 and DAVID6.8 GOEAs whose Benjamini values were less than 0.05 using DAVID's GOTERM_BP_ALL terms for both sets of analyses.

The most notable difference was DAVID6.8 found about four times as many unique GO terms to be significant than were found by DAVID6.7 (1,617 GO terms for DAVID6.8 vs 390 GO terms for DAVID6.7) for all Gjoneska clusters.

Also using the GO DAG downloaded in April 2018, the dcnt for significant GO IDs for DAVID6.8 was smaller (52 median, 296 mean, SD=1,192) than for DAVID6.7 (163 median, 668 mean, SD=1,672).

Effects of the same annotations in DAVID6.8 and GOATOOLS

To examine the effects of using the same annotations in DAVID and GOATOOLS, I ran GOATOOLS GOEAs with annotations downloaded from DAVID6.8 using *Fisher's exact test* and both the *Benjamini-Hochberg* and *Bonferroni* multiple test corrections. GOATOOLS generally found more GO terms than found by DAVID6.8 (Supplemental Table 4).

Tool comparison

The P-values found by all tools had similar statistics overall (Figure 5.8D). DAVID6.7 found GO terms that were ten times broader than other tools and completely missed many specific GO terms (Figure 5.8A). GOstats found GO terms that were almost twice as broad as GOATOOLS and DAVID6.8.

The broad GO terms found by GOstats and DAVID6.7 could sometimes be exceptionally broad and associated with hundreds of genes yielding impractical results. For example, the particularly broad depth-01 term, *cellular process* (GO:0009987) with its over 18,000 descendant GO terms may not be helpful in describing unique properties of a gene set. Also, such broad terms may result in the addition of discovering hundreds of genes that are only associated with broad terms having low information content (Supplemental Figure 4). And finally, including extremely broad GO terms in GOEAs may cause GOEAs to have unacceptably high FDRs which exceed the alpha set by the researcher (Supplemental Figure 1).

GOstats and DAVID6.7 (using the GOTERM_BP_ALL GO set) found more broader GO terms than GOATOOLS and DAVID6.8 (using the new GOTERM_BP_DIRECT GO set). Finding more broader GO terms may be due to GOstats and DAVID6.7 using a variation of propagate_counts to augment the original annotations.

My stochastic simulations show that using propagate_counts can result in greater sensitivity to find truly enriched genes (5.4B) rather than missing them. If using propagate_counts, it may be especially important to remove extremely broad GO terms that are better represented by numerous specific GO terms

prior to the analyses to prevent FDR values from exceeding the alpha set by the researcher.

I chose to not use propagate_counts in GOATOOLS and to use the GOTERM_BP_DIRECT DAVID annotation set for the analyses in this paper to investigate the GOEA results using original unmodified annotations. In actual practice, it may be desirable to run GOEAs trying both the original unmodified annotations and propagate_counts.

Conclusion

The first stochastic simulations failed, meaning that the FDR exceeded the alpha set by the researcher (Supplemental Figure 1). The source of the failures were false positives involving extremely broad GO terms associated with more than one thousand genes for *biological_process* in the mouse annotations. Simulations passed if just 30 broad GO terms out of more than 17,000 total annotated GO terms are removed from the annotations prior to running the GOEAs. Therefore, developers of GOEA tools may want to consider removing even a small number of GO IDs associated with large numbers of genes if the broad GO term may be better represented by numerous annotated more specific descendant GO terms.

Stochastic simulations revealed that augmenting the annotations using *propagate_count* set to "True" to cause parent GO terms to be added to a gene product's annotations resulted in better sensitivity in finding truly enriched results which would otherwise not be found (Figure 5.4B). Smaller study gene sets (4-20 gene products) most dramatically benefited from propagating GO

annotations depending on the percentage of truly enriched gene products in the study sets (Figure 5.4B, panels B3-B5).

Because using any variation of *propagate_counts* comes at the expense of finding more broad terms, developers of GOEA tools should strongly consider pruning selected broad terms that are associated with large numbers of genes and have numerous descendants prior to running GOEAs. Researchers may wish to run a GOEA twice, once with the original annotations and once with the annotations augmented by propagating annotations up through GO parents.

Numerous GO terms, especially large groups of specific GO terms, can be difficult to summarize. GOATOOLS grouping not only makes a single set of GOEA results easier to understand from a systems level, but it also makes it possible to compare GOEA results across multiple tools, species, or experiments even if the GO terms from the various tools or experiments are at different depths.

The GOATOOLS library can help the researcher keep current with rapidly changing ontologies and associations as well as organize and summarize GOEA results. Given Python's popularity among bioinformaticians and data scientists, GOATOOLS fills a significant void while maintaining comparable if not better performance than other tools and libraries that are built using other programming languages.

Chapter 6: Gene product semantic similarity

6.1 Introduction

6.1.1 Functional relationships among genes

A protein can be better understood by comparing it to related proteins. But comparing biological processes and molecular functions between two proteins cannot easily fit into a mathematical form, which can be achieved when comparing sequences and structures of two gene products using alignment programs.

It is valuable to compare proteins using their biological functions rather than only their sequences, because proteins declared to be functionally similar due to their sequence similarity could be wrong 30% of the time [45] [44]. This calls for comparing proteins using functional similarity, which complements and augments analyses using sequence similarity.

Biological knowledge is recorded in research papers using natural language describing hypotheses, discoveries, and the analysis of experimental data. Finding, organizing, and visualizing biological information is most reproducible when performed computationally, but biological knowledge described using natural language does not lend itself to computational methods. Such knowledge can be made more amenable to computational access when stored in ontologies.

Ontologies are structured, objective knowledge representations used throughout the research community, and they contain biological knowledge described using natural language in peer-reviewed research papers. Ontologies allow comparing gene products using descriptive terms scaled to a

genomic level computationally rather than comparing small numbers of proteins one by one.

6.1.2 Gene ontologies

Gene ontologies (GOs) are terms that describe biological processes, molecular functions, or the locations of gene products with reference to the cellular compartment. The purpose of GOs is to represent biological knowledge as it constantly expands and becomes revised. The GOs are widely used among the life sciences community, contain over 45,000 GO terms, and are under continual development with the addition of new terms and annotations of GO terms to gene products from contributions the scientific community [206] [10].

Gene function is described using GO terms through annotations, as shown in Figure 6.1. Annotations are GO terms that are associated with a gene to describe its function. In Figure 6.1 for example, gene A is described using two GO terms, *adaptive immune response* and *germinal center formation*, while gene G is described using one GO term, *adaptive immune response*.

The GO terms are hierarchically related through their storage in a directional acyclic graph (DAG). For example, the GO term *response to interferon-beta* is the child of the more general parent term *response to cytokine*. Unlike tree structures, where all nodes have a single parent, nodes in a DAG can have multiple parents.

Comparing two GO terms is not straightforward due to the likelihood of a GO term having multiple parents in the DAG and the highly skewed nature of the GO DAG, leading to some branches in the GO having a maximum depth of

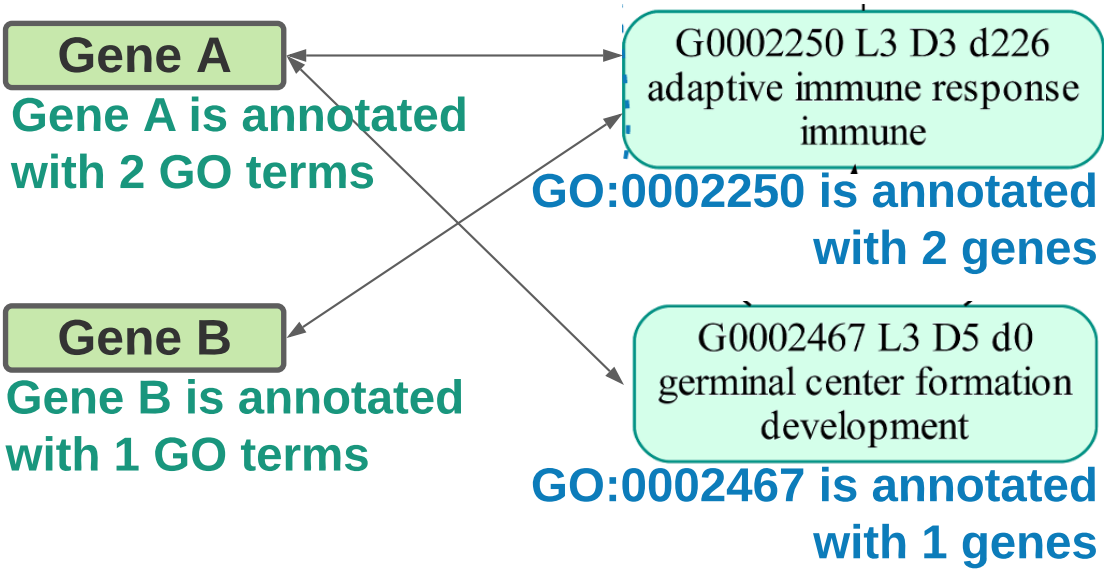

Figure 6.1: **Gene function is described using GO terms through annotations.** Gene A is annotated with two GO terms, *adaptive immune response* and *germinal center formation*. Gene B is annotated with one GO term, *adaptive immune response*.

over 15 from root to leaf, while other branches only have a maximum depth of 2 [106].

6.1.3 Semantic similarity methods for a pair of GO terms

Numerous methods can be used to compare a pair of GO terms (Table 6.1). The semantic similarity equations for three of the methods (Resnik, Lin, and Schlicker) are shown in Table 6.2. Term-based semantic similarities are comparisons between two GO terms using a semantic similarity calculation.

Gene-functional semantic similarities are comparisons between two gene products. Genes and gene products such as proteins are annotated with a set of GO terms (Figure 6.1). Pairs of genes or gene products are compared by

Table 6.1: **A list of various gene ontology semantic similarity methods.** MICA: Most informative common ancestor; DCA: Disjoint common ancestors; IC: Information content; anno: Annotations;

Year	Author	Cite	Method	Techniques	Name
1995	Resnik [177]	4085	Node	MICA	Resnik
1998	Lin [122]	4948	Node	MICA	Lin
1997	Jiang & Conrath [99]	3578	Node	MICA	JC
2006	Schlicker [187]	582	Node	MICA	Relevance
2005	Bodenreider	128	Node	Shared anno	
2005	Couto [39] [38]	118	Node	DCA	GraSM
2007	Wang	758	Hybird	Shared ancestors	G-SESAME
2007	Othman	64	Hybrid	IC/depth/ccnt/dist	
2007	Riensche	25	Node	IC/MICA; shared annos	
2005	Wu	153	Edge	Shared path	
2006	Wu	238	Edge	Shared path	
2005	Yu	79	Edge	Shared path	
2004	Cheng	136	Edge	Shared path	
2008	del Pozo [171]	85	Edge	Shared path	
2008	Pesquita [169]	85			simUI, simGIC
2012	Sanchez [184]	319			
2012	Yang [217]	76			GOsim

comparing the set of GO terms for each gene using semantic similarity measures.

Algorithms for calculating similarities between two GO terms can be summarized as edge-based, node-based, or hybrid, meaning that they use both edge and node properties. Edge-based methods rely on properties of the edges between GO terms, such as the distance between the GO terms and depth of the common ancestors from the top root term. Node-based methods rely on the properties of the GO terms, such as the genes annotated to a GO term and

the number and relationship of ancestor terms above the GO term and descendant terms below the GO term.

Node-based methods of scoring similarity between two terms are considered more accurate with the GO DAG due to its highly skewed nature at all levels [169].

6.1.4 Annotations of GO terms to gene products

Most node-based semantic similarity measures are based on information content (IC), which is determined by counting the instances that a gene product is associated with a GO term. For example, Figure 6.2 shows one gene (gene b) associated with node E, four genes associated with node B (genes a, b, c, and g), and ten genes associated with root node A (genes a through j).

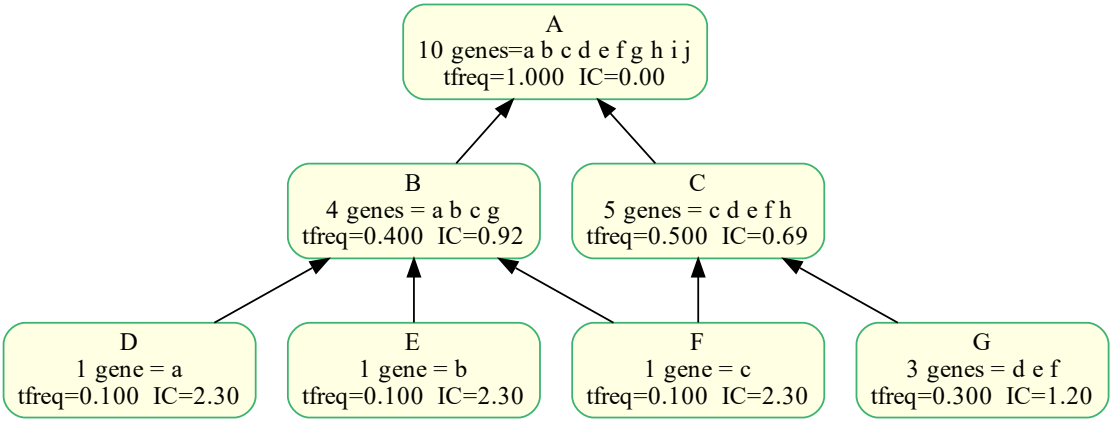

Figure 6.2: **Information content and GO term frequency.** Nodes (GO terms) are represented by boxes. Relationships between the nodes are represented by arrows. The number of genes annotated to a node is shown along with the gene names, ranging from *a* to *j*. There are ten genes in total annotated to the GO terms in this example GO DAG. The frequency of a term is shown as *tfreq* on each node. The IC of a term is indicated with *IC*.

Annotating a gene to one GO term is defined as associating the gene to all the GO term's ancestors. In Figure 6.2, each gene is annotated to one GO term. Gene a is associated with only leaf term D in the annotations, but gene a is also associated with nodes B and A since they are a parent and ancestor of node D.

There are a total of ten genes in the annotation (genes a, b, c, d, e, f, g, h, i, and j), which are all shown to be associated with root node A. The frequency with which genes are seen on a node regards the number of genes annotated to the node divided by the total number of genes in the entire annotation. In Figure 6.2, node B has a frequency of 0.400 ("tfreq=0.400" on node B) since it is associated with 4 out of 10 total genes. The root node is always associated with all the genes and has a frequency of 1.00.

6.1.5 Information content

Information content (IC) is defined as the negative log10 or negative natural log of the frequency that a GO term is annotated to a gene product. The IC is indicated in the node boxes in the Figure 6.2 with "IC=". Terms associated with high numbers of genes (Figure 6.2 node A) are considered to have less information than terms associated with low numbers of genes (e.g., Figure 6.2 nodes D, E, F, and G).

As an analogy to liken people to genes and GO terms, consider the geographic locations Earth and Philadelphia. If all people living on Earth are associated with Earth and all people living in Philadelphia with Philadelphia, then saying that you live on Earth is less informative than saying you live in Philadelphia, because Earth has so many more people associated.

6.2 GO term semantic similarity

The definitions of GO term semantic similarity values between GO_1 and GO_2 are defined in Table 6.2. The gene functional semantic similarity method used in this thesis is Schlicker's relevance score augmented with Yang's random walk constraint (RWC).

Table 6.2: **Semantic similarity equations.** Equations for information content (IC) and three GO term semantic similarity measures: Resnik, Lin, and Shlicker's relevance.

Equation	Name
$IC(GO) = -log\, p(GO)$	IC
$sim_{Resnick}(GO_1, GO_2) = log\, p_{MICA}(GO_1, GO_2)$	Resnick [177]
$sim_{Lin}(GO_1, GO_2) = \frac{2 \times log\, p_{MICA}(GO_1, GO_2)}{log\, p(GO_1) + log\, p(GO_2)}$	Lin [122]
$sim_{Lin}(GO_1, GO_2) = sim_{Lin}(GO_1, GO_2) \times (1 - p_{MICA}(GO_1, GO_2))$	Schlicker [187]

6.2.1 Resnik's semantic similarity score

Resnik's score is the negative log of the IC of most informative common ancestor (MICA) of a pair of GO terms (Table 6.2), with values ranging from zero and upwards. In Figure 6.3, the light grey node pair H and I and the medium grey node pair D and E share common ancestors A and B. Node B is more informative than A because B's IC of 0.400 is higher than A's IC of zero, and thus B is the MICA.

Node pair D and E are more similar to one another than H and I, because MICA B is closer to both D and E than to H and I. The MICA B separates H and I by a great distance, so they are the least similar pair in Figure 6.3. But the Resnik score, 0.400, is the same for both node pairs D and E as well as H and I, because both pairs of GO terms have the same MICA, B. Thus Resnik's score is thus not sufficient to choose as a measure.

6.2.2 Lin's semantic similarity score

Unlike Resnik's, Lin's score uses the IC from each node in the pair being compared (Table 6.2). Lin's score is a ratio, so it ranges from zero to one. Lin's method accurately considers nodes D and E (medium gray) more similar than H and I (light gray), with scores of .60 (more similar) and 0.23 (less similar) respectively.

The node pair F and G (dark gray) is more similar than D and E (medium gray) because their MICA (C) is more specific than the MICA (B) for D and E. Lin's score does not account for the broadness of the MICA, so Lin's score will be the same (0.60) for both D and E and for pair F and G. Lin's score is not accurate in this example.

6.2.3 Schlicker's relevance semantic similarity score

Schlicker's relevance score solves the issue of the broadness of a MICA by factoring the term frequency of the MICA into the score (Table 6.2). The relevance score reduces the Lin score by one minus the MICA's term frequency.

Lin's score is 0.60 for both pairs D and E (medium gray) as well as F and G (dark gray), incorrectly indicating that both pairs are equally similar. The Lin

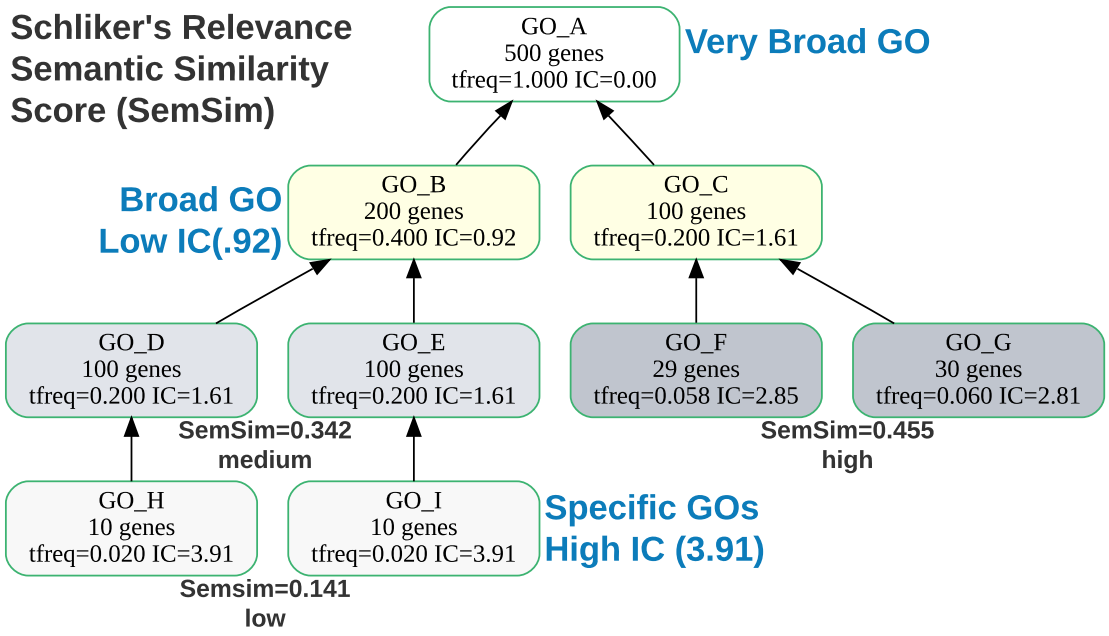

Figure 6.3: **GO term semantic similarity methods: Resnik, Lin, and Schlicker.** GO pairs are shaded in grey, with the darkest pair, F and G, being most similar to each other and the lightest pair, H and I, being the least similar to each other. Resnik's score is the same for pair H and I and pair D and E, and it does not reflect that H and I are less similar than D and E. Nodes are represented by boxes. Relationships between the nodes are represented by arrows. The number of genes annotated to a node is shown along with the gene names, which range from *a* to *i*. Semantic similarity scores are shown between E and B, D and B, F and C, and G and C. There are five-hundred genes in total annotated to the GO terms in this GO DAG. The frequency of a term is shown as *tfreq*. The IC of a term is indicated with *IC*.

score for D and E is reduced by 40% to acquire the relevance score of 0.342 (Figure 6.3, medium gray). The Lin score for F and G is reduced by only 20% to 0.455 (Figure 6.3, dark gray), reflecting that pair F and G are more similar to one another than D and E since their MICA C is more specific than MICA B.

6.2.4 Yang's method

All similarity measures before Yang's only used ancestor terms above the pair of GO terms being compared, disregarding any descendant terms below. Yang et al. addressed the effect of the descendants below the GO pair and its importance. Yang's random walk constraint (RWC) method was introduced in 2012 [217] and implemented in Java (https://github.com/pwac092/gossto) as an open-source software project in 2014 [28].

The GO similarity measures before Yang also fail to consider the GO terms not yet added to the GO DAG, meaning that the gene annotations were added to broader GO terms but not yet assigned to specific low-level GO terms since detailed experiments were not yet performed.

Yang's method is an add-on to popular semantic similarity measures such as Resnik [177], Lin [122], and Schlicker's relevance score [187], which only consider ancestor terms when assessing similarity.

Figures 6.4 and 6.5 are used to explain why the similarity between the node pair D and E (yellow) differs from that between F and G (cyan) when considering the descendant terms but is the same when only considering the ancestor terms.

The number of genes annotated to a node is shown in the node boxes in both figures. In this example, the genes directly annotated to a node are not repeated; for example, in Figure 6.4, node H is annotated with 10 genes and node I is annotated with 10 more different genes.

The equations in nodes D, E, F, and G show the total of genes annotated to a node: total annotated = directly annotated + annotated through descendants.

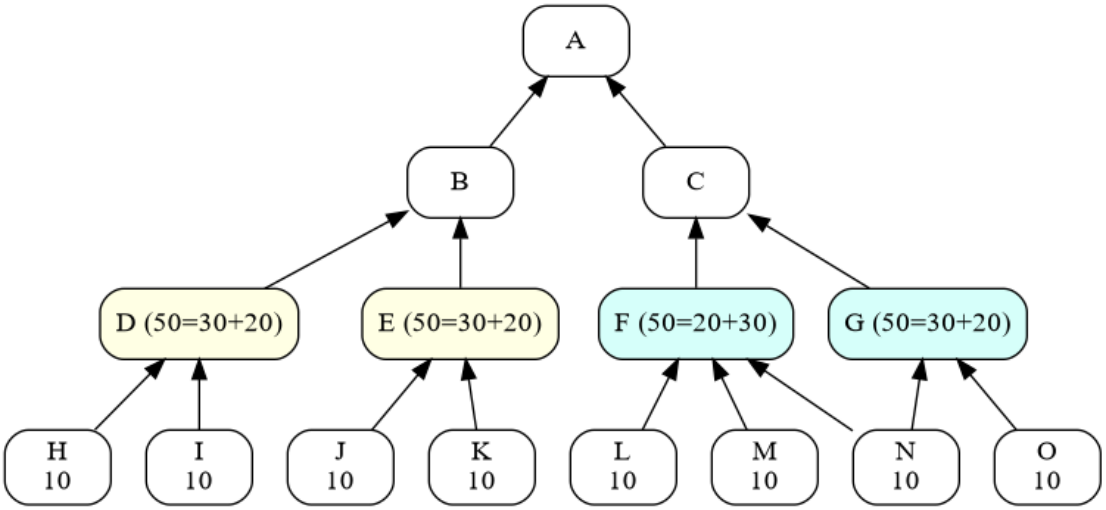

Figure 6.4: **GO term nodes are more similar to one another if they share descendants.** Nodes F and G (cyan) are more similar to one another than D and E (yellow), even though the hierarchy above both pairs of nodes is equivalent. Nodes D and E share no descendants, while F and G share N, making F and G more similar to one another than D and E. Nodes are represented by boxes. Relationships between the nodes are represented by arrows. The number of genes annotated to a node is shown for the lower-level nodes. This figure was adapted with permission from Figure 1a in Yang et al. [217].

For example, in Figure 6.4, the equation in node D is 50=30+20, meaning 50 total genes are annotated to node D, with 30 direct annotations on node D and 20 annotations inherited from the 10 annotations on node H plus the 10 annotations on node I. The 30 direct annotations are unique and not repeated on any other nodes for this example.

Shared children increase the similarity of a GO pair

The GO DAG structure underneath the pair of terms being compared affects the similarity of the GO terms. Yang describes how to use Figure 6.4 [217] as

follows: Node pair D and E would have the same similarity score as node pair F and G if only their ancestors (A, B, and C) were considered, because the ontology structure above the D-E and F-G is equivalent, and nodes D, E, F, and G are all annotated with 50 genes.

Nodes F and G would be declared more similar than nodes D and E when considering the descendant nodes, because F and G share a child (N) while D and E share no children.

Annotations inherited from a GO pair increase its similarity

The second reason descendant terms should be used to determine the similarity between two GO terms is to account for uncertainty, which considers GO terms not yet added to the DAG.

In Figure 6.5, nodes D and E (yellow) are less similar to one another than F and G (cyan). Comparing D and E is like comparing a broad term D with the specific term E, while comparing F and G is like comparing two more specific terms.

Node F is considered more specific than D since it inherits all its annotations from its children, while D has 18 direct annotations and only inherits 2 annotations from its children.

The annotations to D are described as uncertain by Yang since they will likely be moved to a yet-to-be determined, more specific term after obtaining sufficient experimental evidence.

Yang et al. showed that adding their method to others consistently improved similarity comparisons when comparing data from sequence

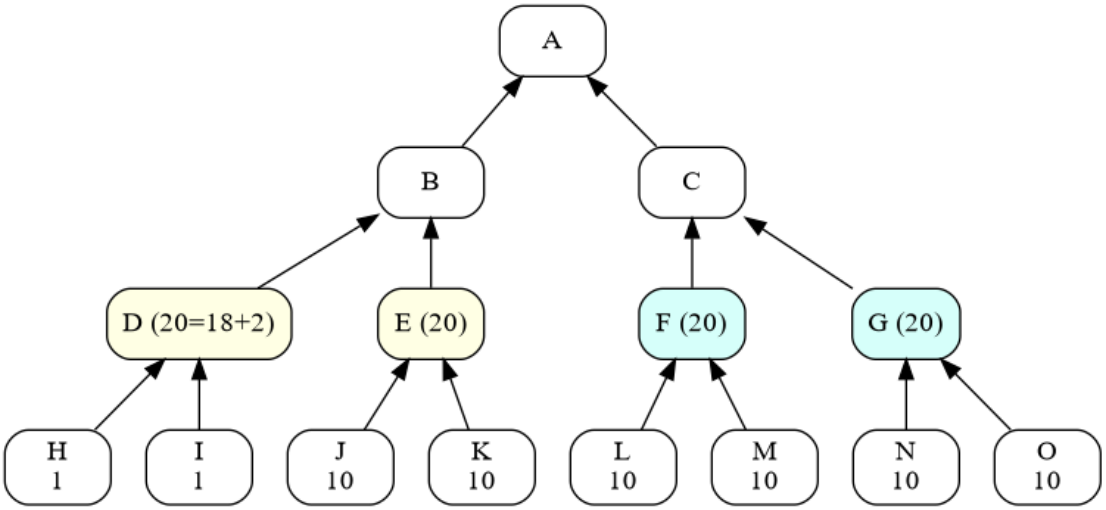

Figure 6.5: **Direct versus inherited annotations affect a term's specificity.** Node D is a broader term than node F even though they are both annotated with 20 genes, because 18 genes are annotated directly to D and all genes annotated to node F are annotated through its more specific descendants, L and M. Nodes F and G are more similar to one another than D and E even though the hierarchy above the nodes is equivalent, because node F has less uncertainty than node D. Relationships between the nodes are represented by arrows and nodes are represented by boxes. The number of genes annotated to a node is shown for the lower level nodes. This figure was adapted with permission from Figure 1b in Yang et al. [217].

similarity comparisons, gene expression experiments, and protein-protein interactions.

6.2.5 Methods not chosen

Other methods not chosen for use in this thesis include those of Peng and Mazandu. Peng's method was initially appealing due to several published papers discussing semantic similarity and proposing novel methods along with providing links to tools for their implementation.

Peng's most recent paper published in 2019 discusses improving the clustering of single-cell RNA-seq data by combining GO with deep neural networks [166]. Peng is shown as a visiting Associate Professor at Harvard Medical School in Clemens R. Scherzer, M.D.'s Neurogenomics Laboratory https://www.scherzerlaboratory.org/members/index.html (accessed October 10, 2019).

Peng's tools include InteGO [162], InteGO2 [163], NETwork-based SIMilarity measure (NETSIM) [165], NETSIM2 [167], and a web tool for tissue-specific gene ontology enrichment (TSGOE) [164].

Compiled Java code (but no source code) is available for NETSIM at `http://www.msu.edu/~jinchen/NETSIM`. It was not clear where to find NETSIM2 since no link was provided in the manuscript. A link was provided to the TSGOE web tool in the manuscript `http://120.77.47.2:5678`, but the URL could not be retrieved when attempting to visit the site.

The original link provided in InteGO [162] and the InteGO2 [163] manuscripts, `http://mlg.hit.edu.cn:8089/`, could not be reached, and the new link listed in the erratum for InteGO2, `https://mlg.hit.edu.cn/InteGO2`, was also inaccessible. The Harbin Institute of Technology website, `https://mlg.hit.edu.cn/` was also unavailable. Trying the English version, `https://en.hit.edu.cn/`, also yielded no results. A Google search on Oct 14, 2019 revealed the following: "Harbin Institute of Technology (HIT) was founded in 1920. From its beginning, HIT has received preferential support from China's Central Government."

Mazandu's method and tool, A-DaGO-Fun, was initially appealing since it was written in Python and is newer than Yang's tool [133]. In addition, Mazandu has written two reviews of semantic similarity measures [134] [135]. The A-DaGO-Fun code, however, only runs in Python2, which is being replaced by Python3 and is scheduled for its end of life in 2020. And the popular Python code linter, called pylint, revealed too many code issues including numerous "too many nested blocks," "too many branches," and others.

A code linter is a static analysis tool which identifies programming bugs and stylistic errors that may not be readily identified by running dynamic tests or by compilers for compiled code written in languages such as C and C++ [100]. A dryer machine lint trap collects small bits of fiber and fluff, preventing a fire from a dryer pipe clogged with lint, while a code linter similarly finds small bits of code before they could potentially cause greater problems when the code is deployed for public use.

6.3 Gene functional semantic similarity

Gene products can be functionally related in a number of ways, such as through direct protein-protein interactions (PPIs). Alternatively, two proteins can be in the same molecular complex but not be physically touching each other. In addition, two functionally related proteins can be present in the same biological pathway but not have any PPI or be in the same molecular complex. Functionally related genes can even be in the same biological pathway but not

be in the same cellular compartment for pathways that extend over multiple cellular locations.

Gene functional similarity is used for gene function prediction, protein-protein interaction prediction, disease gene prioritization, gene clustering, gene network analysis, gene association visualization, and missing value imputation.

6.3.1 Semantic similarity for a pair of gene products

To compare a pair of gene products, all GO terms annotated to each gene must first be compared to one another using pairwise GO term semantic similarity calculations as shown in Figure 6.6. Each gene product in a pair has a number of associated GO terms; in Figure 6.6 for example, gene A is annotated with GOa through GOd, while gene B is annotated with GOx through GOz. Each cell in the table contains the semantic similarity score for one pair of GO terms. For example, the lower right-hand cell is the semantic similarity (0.455) score for GOd and GOz. Each cell in Figure 6.6 was filled by calculating the pairwise GO term semantic similarity measures for a GO in the row and a GO in the column of the cell.

The next step is to obtain one score representing the extent of the similarity between gene A and gene B. There are numerous methods to obtain a value. Using an average value would cause gene products that were similar in some functions but not in others to be inaccurately assessed as having little similarity. Choosing the maximum gene-to-gene semantic similarity value determines whether two genes have strong similarity or no similarity, and thus represents the function used in this thesis. In Figure 6.6, the maximum GO pair

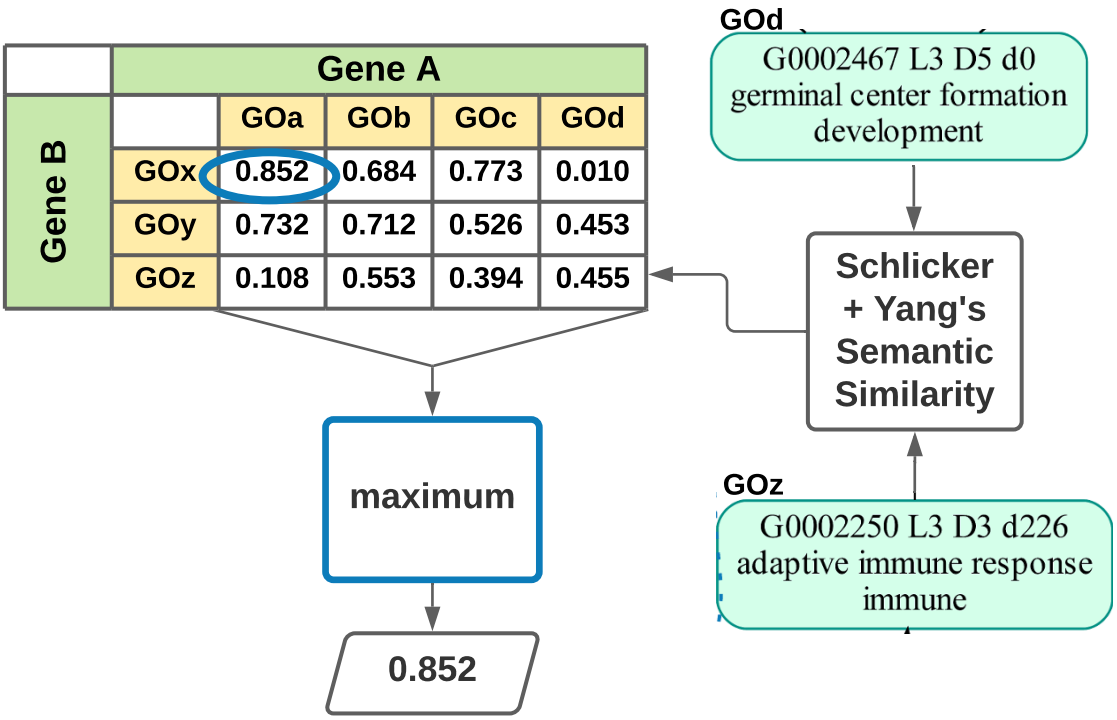

Figure 6.6: **Compare two genes by calculating pairwise similarity of all GO terms.**

semantic similarity score is 0.852, so this value is used to represent the similarity of the gene products.

6.3.2 Comparing many gene products from one cluster

To determine whether a cluster has many genes that are functionally similar, the genes in a cluster are placed both in the rows and columns of the gene functional semantic similarity matrix, as shown in Figure 6.7. Only the lower triangle of the gene functional semantic similarity is needed, because all diagonal values are always comparing a gene to itself. The value is thus always 1.0, which gives no new information about the similarity between the

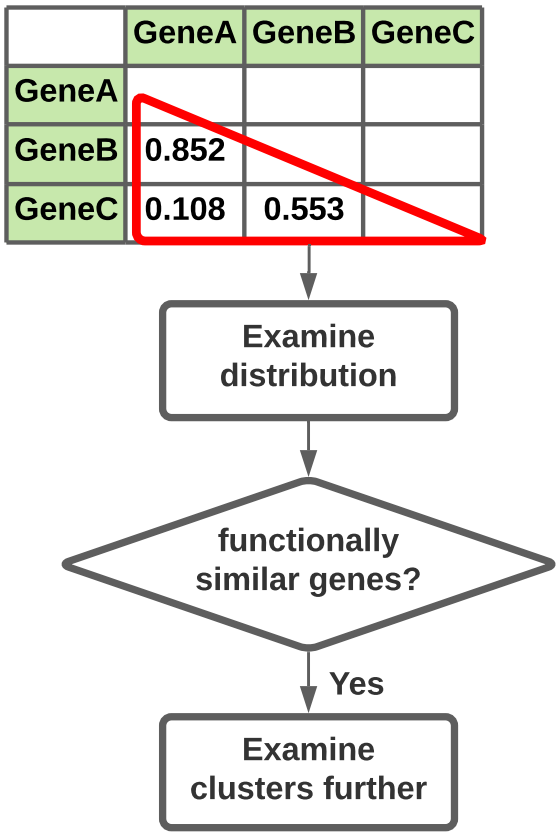

Figure 6.7: **Flowchart for comparing the similarity of cluster genes.**

The gene function semantic similarity values are stored in a lower triangular matrix (outlined in red).

cluster genes. The upper triangle is not needed because its values are a duplicate of the lower triangle.

6.3.3 Flow to determine the similarity of cluster genes

Figure 6.8 shows the steps to determine whether a cluster has more genes that are functionally similar to one another than would occur by random chance.

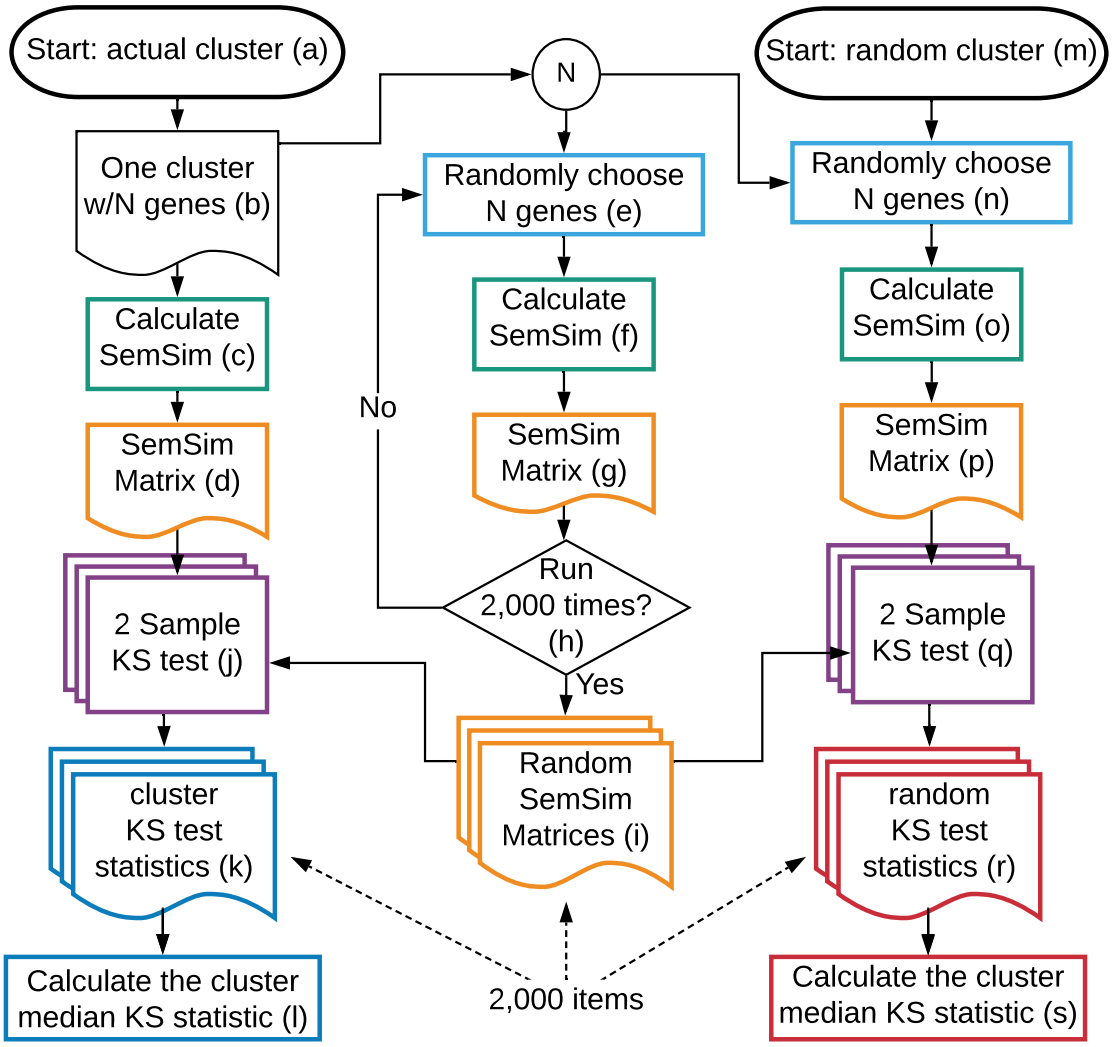

Figure 6.8: **Flow diagram for two-sampled KS-statistics test to determine which clusters have many genes with similar functions.** (N: the number of genes in a cluster; SemSim: semantic similarity; KS test: two sample Kolmogorov-Smirnov statistical tests)

There are 3 columns in Figure 6.8. The left-most column is the branch in the statistics flow for an actual cluster, which has N genes, while the right-most column is for a stochastically generated cluster of N genes. The middle column shows the flow to generate a background set of 2,000 randomly generated

clusters of N genes. A stochastic cluster is generated by choosing N genes randomly from the chromosome that contains the current cluster.

The gene functional semantic similarity (SemSim) scores are calculated as described in the previous section for the current gene cluster and for every random gene cluster (Figure 6.8, left column), which includes the 2,000 background clusters plus the 1 random cluster (Figure 6.8, green boxes c, f, and o). The triangular matrices' SemSim values (Figure 6.7) appear in Figure 6.8 as orange data boxes.

The two-sampled Kolmogorov-Smirnov test determines whether the cluster distribution of SemSim values is similar to the random distribution of SemSim values (Figure 6.8, purple box j). Each run of a KS-test (purple box j) takes the cluster SemSim (orange box, d) and one of the background stochastic SemSims (orange box, i) as input to produce one KS-test statistic (blue box k).

The two-sample KS-test is repeated 2,000 times to compare the current cluster (orange d) with each of the 2,000 stochastic background SemSim matrices (orange i) to generate a total of 2,000 KS test statistics for the current cluster (blue boxes k). This process is repeated for the random cluster (elipse m) to generate a set of 2,000 KS test statistics (red boxes r).

6.3.4 Visualize the KS statistics for each cluster

Figure 6.9 shows the distribution of the KS statistics for a sample of 10 clusters from over 100 total clusters. Each blue shape in the violin plot (Figure 6.9) represents 2,000 KS statistics for a cluster. The 2,000 KS statistics were previously shown in the flow diagram in Figure 6.8 as blue data shapes (k). Each red shape next to a blue shape in the violin plot represents 2,000 KS

statistics for a randomly generated cluster with the same number of genes as the actual cluster (Figure 6.8, red data shapes r).

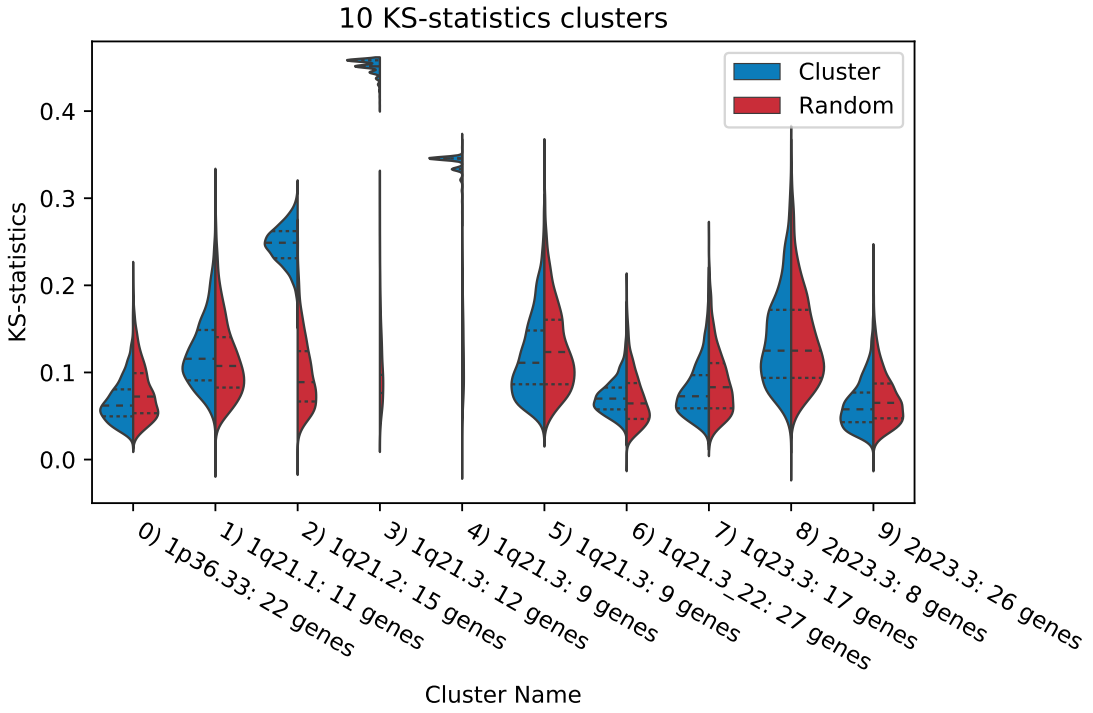

Figure 6.9: **Cluster versus random KS-test statistics for 10 clusters for 10 of over 100 clusters.** There are two sets (red and blue) of 2,000 KS statistics for each of the 10 clusters shown. A single violin plot represents one cluster. The blue part on the left-hand side of each violin plot shows the distribution of the KS statistics for each cluster compared to 2,000 randomly generated clusters. The red part on the right-hand side of each violin plot shows the distribution of the KS statistics for a randomly generated cluster compared to 2,000 randomly generated clusters.

Cluster 0 has a distribution of 2,000 KS statistics approximately mirroring that of the randomly generated cluster (red), meaning that the genes in cluster 0 are no more similar to one another than a set of randomly chosen genes. Cluster 2 has a KS statistic distribution (blue) that is dramatically different

than the randomly generated cluster (red), meaning that many genes in cluster 2 are functionally similar to one another.

To obtain a p-value for each cluster, I used the Mann-Whitney U test on each cluster to compare the set of 2,000 KS-statistics for the cluster with the set of 2,000 random KS-statistics, because the KS-statistic values in each set are continuous but not normally distributed, and the two sets are independent groups.

6.3.5 Visualize all clusters against random clusters

Figure 6.10 shows a second visualization of all 100-plus clusters.

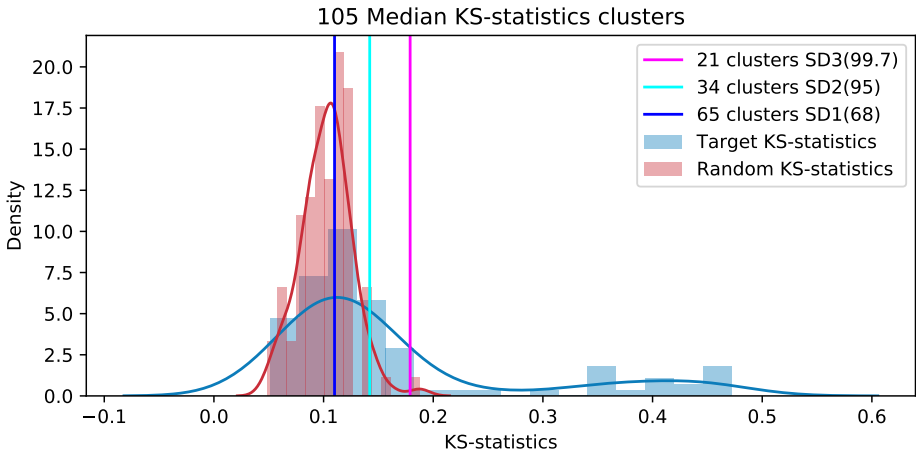

Figure 6.10: **Statistics for gene functional semantic similarity for all clusters.** The orange bars are the KS statistics for each of the 105 protein-coding clusters. The blue bars are stochastically generated KS statistics for 105 clusters. There are 34 orange clusters to the right of the cyan line, whose KS statistics are statistically significant compared to the randomly generated clusters.

I calculated the median value of each set of 2,000 KS statistics, which resulted in 100-plus cluster medians (light blue) and 100-plus random medians

(light red). The medians were previously shown in the flow diagram in Figure 6.8 as a blue box (l) and a red box (s). There is one KS-test median per cluster. In the histogram (Figure 6.10), the light blue bars to the right of the vertical 95th percentile line (cyan line) show the median of the KS statistics for clusters that are more likely to have genes that are functionally related than due to chance alone.

6.4 Overall results

Genes that are in the same gene family and co-located linearly have similar functions, as expected; however, there is also frequently a striking functional resemblance of genes near but not a member of neighboring gene families.

There are 33 out of 105 clusters with genes that are more similar to one another than would be found by random chance, which are shown as all blue clusters to the right of the cyan line in Figure 6.10. The genes that are not in the same family as a neighboring gene family but exhibit similar functionality appear in Table 6.3. All but one of the 33 clusters to the right of the cyan 95th percentile line in Figure 6.10 are statistically significant as calculated by the Mann-Whitney U test. The one cluster (9q32) past the 95th percentile line that was not statistically significant was close to the cyan line and had no interesting genes.

Table 6.3: **Interesting genes that share common functionality with genes from a nearby gene family for clusters containing 10 or more genes.** I: the index of the cluster; Start bp: the starting base pair (bp) location on the chromosome; Length: the length of the cluster (bp); N: the number of genes in the cluster; Gene symbols: the gene symbol of interesting genes in the cluster.

I	Location	*p*-value	Start bp	Length	N	Gene symbols
1	1q21.2	0.00000E+00	149,701,425	235,473	18	MTMR1, SF3B4, SV2A
2	1q21.3	0.00000E+00	152,510,844	208,678	13	C1orf68
3	1q21.3	0.00000E+00	152,758,030	150,533	10	IVL
4	2q31.1	0.00000E+00	176,002,901	266,494	12	LNPK
5	2q37.1	0.00000E+00	233,566,789	207,233	12	
6	3p21.31	3.16815E-281	50,088,908	270,600	19	GNAT1, SEMA3B, SEMA3F, SLC38A3
7	3p21.2	8.02810E-41	51,941,941	114,504	9	
8	4p16.1	0.00000E+00	9,210,657	233,877	20	
9	5q31.1	8.17659E-145	132,658,173	393,797	12	
10	5q31.3	0.00000E+00	140,547,666	418,496	29	CD14
11	5q31.3	4.62295E-68	141,051,376	437,865	40	
12	6p22.2	0.00000E+00	26,017,812	106,333	14	HFE
13	6p22.2	0.00000E+00	26,156,331	129,168	19	
14	6p22.1	0.00000E+00	27,807,931	149,310	18	H1-5
15	6p21.33	2.46559E-176	43,427,538	260,274	12	C2, C4A, C4B, CFB, LTA, MICB, NFKBIL1
16	7p15.2	0.00000E+00	27,096,006	146,539	11	
17	7q36.1	3.98533E-102	150,978,314	107,517	9	
18	8q24.3	0.00000E+00	7,784,165	193,820	11	ARC, LYNX1, PSCA
19	9p13.3	1.77295E-43	34,590,386	139,152	12	CCL27, CNTFR
20	11q12.3	2.71481E-08	62,646,726	81,347	12	
21	12q13.12	0.00000E+00	48,957,526	173,995	10	
22	12q13.13	0.00000E+00	53,938,792	78,060	7	
23	14q11.2	1.35782E-254	20,305,994	204,478	9	
24	14q11.2	2.48661E-60	23,010,201	175,083	10	
25	16q13	0.00000E+00	56,589,355	93,115	10	
26	17q21.2	0.00000E+00	41,027,202	96,965	13	
27	17q21.2	0.00000E+00	41,226,648	89,047	10	
28	18q21.1	0.00000E+00	46,917,602	331,922	10	PIAS2
29	19q13.33	9.82854E-288	49,017,090	68,050	9	LHB, NTF4
30	20q13.33	1.22974E-33	63,959,435	147,396	10	
31	21q22.11	0.00000E+00	30,372,239	270,625	21	
32	21q22.3	0.00000E+00	44,540,195	171,385	17	
33	Xq26.3	0.00000E+00	135,708,398	266,199	12	

6.5 Example cluster: 1q21.3

This section contains an example a cluster with interesting genes, located at 1q21.3. Clusters 2 and 3 in Table 6.3 are concatenated for this example since the two clusters are so close together.

The protein-coding gene cluster in 1q21.3 is about 400 kbp long. The major gene family in this cluster is "late cornified envelope proteins," which is also the name of the eighteen genes in the gene family. Genes in the same family will often but not always start with the same gene symbol prefix, which in this case is LCE. Gene families are hierarchical, and the root gene family for the LCE is the gene family "epidermal differentiation complex," which contains 63 genes, including all 18 LCE genes.

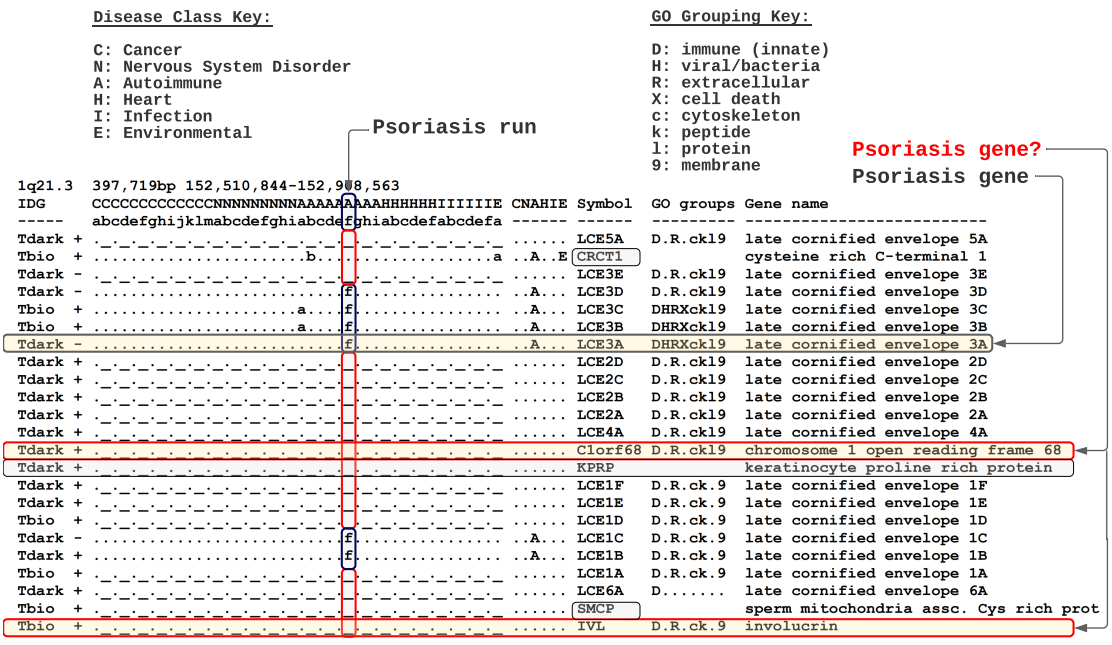

Figure 6.11: **Cluster 1q21.3 has psoriasis, rheumatoid arthritis, asthma, and obesity**

The gene involucrin (IVL) is a member of the gene family "cornified envelope precursor family," which shares the same gene family root as the LCE family "epidermal differentiation complex."

The genes "cysteine rich C-terminal 1" (CRCT1), "chromosome 1 open reading frame 68" (C1orf68), "keratinocyte proline rich protein" (KPRP), and "sperm mitochondria associated cysteine rich protein" (SMCP) are not annotated to any gene families.

The GO group ASCII art patterns are enriched with innate immunity (D), extracellular (R), cell death (X), cytoskeleton (c), peptide (k), protein (l), and membrane (9).

C1orf68 should perhaps also be considered for membership in the LCE gene family since its gene functional profile is similar to that of the LCE genes.

The diseases in the cluster include Rheumatoid Arthritis (Aa), Asthma (Ab), Psoriasis (Af), and Obesity (Ea). There is a disease run across multiple genes in this cluster for Psoriasis, and the genes that break the disease run for Psoriasis are not annotated with any diseases. Genes like IVL and C1orf68 should perhaps be investigated regarding their role in Psoriasis.

6.5.1 Psoriasis

Psoriasis is a chronic inflammatory disease that presents on the skin as red patches covered with silvery scales. It affects men and women equally, more adults than children, and is observed in about 3% of the U.S. population. Psoriasis is associated with numerous chronic comorbidities, including inflammatory arthritis; heart diseases such as heart attack, stroke, and peripheral vascular disease; psychological illnesses such as anxiety and

depression; and inflammatory bowel diseases such as ulcerative colitis and Crohn's disease [9].

The pathophysiology of psoriasis is thought to originate from excessive activity of the adaptive immune system. Activated myeloid dendritic cells ingest antigens near the surface of the skin and then migrate to the lymph nodes, where they invoke adaptive immunity by secreting excessive amounts of the interleukins (IL) IL-12 and IL-23. These interleukins cause naive T cells to differentiate into T-helper cells, which secrete cytokines such as tumor necrosis factor alpha (TNF-alpha), interferon gamma (IFN-gamma), and the interleukins IL-22 and IL-17. The cytokines released by the T helper cells increase signal transduction in keratinocytes, which originate from the innermost layer of the skin and rise to the outermost layer, where they die and become flat, nucleus free, and strong.

6.5.2 Involucrin (IVL)

The outermost tough, strong layer of the skin is called the cornified layer and contains involucrin. When a keratinocyte completes its rise through the skin layers and dies, its plasma membrane is replaced by the cornified envelope, which is a hard, resilient covering. The cornified envelope is composed of crosslinked proteins that include involucrin, envoplakin, and periplakin, and it is surrounded by a lipid envelope that interfaces with the environment [27].

6.5.3 Chromosome 1 open reading frame 68 (C1orf68)

While almost 1,000 research papers in PubMed discuss involucrin, only four appear in a search for "C1orf68." Searching PubMed using synonyms for

C1orf68 (LEP7 and xp32) resulted in two results for “xp32” and none for “LEP7.”

The gene C1orf68 should be of considerable interest to researchers, because despite its scant presence in the literature, when it does appear, it stands out.

For example, in their 2014 study, Edqvist et al. found that C1orf68 was the top gene out of 417 whose protein products were elevated for expression in skin versus non-skin tissues [54], meaning that C1orf68 showed the most specificity regarding skin versus other tissues.

In 2019, Szél et al. completed a proteomic study comparing skin from individuals without psoriasis as well as the lesional and non-lesional skin from individuals afflicted with psoriasis. They found that XP32 (C1orf68) was deferentially expressed between both the skin of healthy individuals compared to the lesional skin of psoriasis patients as well as between the skin of healthy individuals compared to the non-lesional skin of psoriasis patients. As a result, Szél believes that XP32 (C1orf68) may promote the maintenance of the non-lesional state [204] and that studying XP32 (C1orf68) may help researchers understand how psoriatic lesions form and how the non-lesional skin of those afflicted with psoriasis maintain that state.

In 2009, Chen et al. studied the 20S proteasome inhibition. Proteasomes contribute to degrading important regulatory proteins without the help of other molecules and without needing ATP. Proteasome inhibitors proven to have minimal or no toxicity provide a tool for fighting cancers. Chen et al. discovered that XP32 (C1orf68) showed a substantial ability to inhibit the 20S proteosome [31].

These discoveries about C1orf68 make it a compelling subject for further study.

6.5.4 Summary of cluster 1q21.3

All genes in this cluster are Tdark or Tbio according to the IDG, meaning that they are understudied. Psoriasis is associated with about 500 genes in the entire genome, and there are about 120 well studied (55 Tclin and 114 Tchem) psoriasis genes. The drugged psoriasis genes (Tclin) consist of kinases, enzymes, ion channels, GPCRs, and nuclear receptors.

All the genes are experimentally well accessible and have strong genetic evidence, according to Stoeger et al. [198]. Nelson et al. found that targets with strong genetic evidence could double the success rate of a drug during clinical development [147].

In 2008, a genome-wide association study of psoriasis identified the late cornified envelope (LCE) gene clusters as a factor in susceptibility to psoriasis [123]. In 2009, researchers confirmed that the deletion of genes LCE3B and LCE3C in the LCE family was a factor for psoriasis in people of European decent [34]. In 2011, Chinese researchers found the deletion of LCE3B and LCE3C to be factors for susceptibility to psoriasis in the Chinese population [118].

The deletion of genes LCE3B and LCE3C may affect the repair of the epidermal barrier when responding to a skin injury [86]. The poorly repaired skin may allow environmental antigens to penetrate the outermost layer of the epidermis called the stratum corneum, thereby activating the innate immune system and resulting in greater susceptibility to psoriasis.

Chapter 7: Conclusion

7.1 The research question

The research question of this thesis is, "Do genes physically clustered next to other genes linearly on the same chromosome share the same functionality, even if the gene is not in the same gene family as its neighbors?" Although it has been widely accepted since before the 1950s that genes next to each other in bacteria are related in function, it is not widely believed that this is true in eukaryotes, such as humans.

7.2 Methods summary

To answer the research question, first I downloaded the gene families associated with each gene from the Human Genome Organization (HUGO) Gene Nomenclature Committee (HGNC) [24]. I then calculated the gene functional semantic similarity scores between each gene compared to every other gene in a cluster using gene ontology (GO) terms annotated to each gene. Both the GO data and annotations are downloaded from the Gene Ontology Consortium's website [206]. I then created Kolmogorov-Smirnov (KS) statistic values [89] to obtain a distribution of semantic similarity scores for all genes in a cluster and compared the distribution of a set of KS values for a gene cluster KS values for a randomly created gene cluster. The genes in a cluster are considered similar if the cluster distribution and random distribution were statistically significant using a p-value cutoff of 0.05.

To visualize the pattern of gene functional similarity between genes in a cluster, I created a novel method of grouping GO terms, which is described in

my first first-author peer-reviewed publication "GOATOOLS: A Python library for Gene Ontology analyses" [106], published in the journal "Scientific Reports."

It was necessary to create a new GO grouping method due to disadvantages of existing methods. The first disadvantage is that their results are graphical, which requires too much space to simultaneously visualize more than one or two clusters. A more compact, non-graphical format would improve viewing and comparing the GO grouping results of over one hundred clusters in this thesis.

A second disadvantage of existing grouping methods is that they are fully automatic and do not allow the researcher to control grouping choices. GO terms often have multiple parents with similar information content, so it follows that they can be grouped into either hierarchy of the multiple parents. The research question may focus on one GO parent more than another in existing grouping methods, but there was no method for the researcher to specify that one group is more relevant to the question than the other without to my GO grouping method.

7.3 Significance and implications

The KS statistics on the gene functional semantic similarity data supported that gene functions were similar among many genes in many clusters.

The patterns in the GO grouping visualization supported that members of the same gene family frequently showed similar gene functions. They also showed that genes not in the same gene family but physically next to a group

of genes in the same family frequently showed the same gene functional semantic similarity as the genes of the neighboring family.

One implication may be that genes not listed in the same gene family as their neighbors may need to be considered in the same gene family. Another implication is that the identified genes may be involved in the same signaling pathways and physiological processes and warrant further study, especially for genes identified as understudied by the Illuminating the Druggable Genome (IDG) group [180].

7.4 Contributions

My contributions include a method to compare gene function across many genes and clusters; answering open questions regarding gene ontology enrichment analyses (GOEA); an improved PubMed literature search experience augmented from the command-line; and data to support that genes frequently have the same functions as neighbors in a different gene family.

I have also found bugs in the code that creates the annotations from the Gene Ontology Consortium, including one showstopper bug where biological data were missing from an annotation file.

7.4.1 GO Grouping

To compare gene function across many genes and clusters, I created a novel GO grouping method and accompanying GO grouping ASCII art for gene functionality visualization. To create the GO grouping, improved GO plotting

was needed. The GOA TOOLs community has reacted favorably to the highly configurable GO plotting.

7.4.2 GOEA parameter effects

Previously open questions for GOEAs included "How does the number of genes in a study affect GOEAs?" and "Should the parameter, propagate counts, be on or off?" Designing, building, and visualizing the set of 100,000 stochastic GOEA simulations resulted in answering these questions.

The simulation figure was challenging due to the significant amount of information to convey: gene set size; expected true positives; false discovery rate; sensitivity; and specificity. In their 1995 seminal paper [15], statisticians Benjamini and Hochberg's figure provided the inspiration needed to create the layout for my figure to succinctly show the full set of information.

The answers revealed by the simulations were important to the research community since gene set size and the propagate counts parameter cause significant variations in the GOEA analyses results.

7.4.3 Literature search improvements

It was not widely believed that genes next to neighboring genes can frequently have similar functionality in eukaryotes such as humans. This belief was mirrored in the results of my first literature search, performed primarily with Google Scholar, which resulted in mostly older, highly cited papers which discussed how genes in the same family frequently have similar functions.

The first Google Scholar literature search was not reproducible, and downloading search results in bulk by copy-and-pasting from the screen was

challenging and hindered finding the latest high-quality research concerning gene functional similarity.

The next literature search using PubMed, augmented from the command-line using my open-source project pmidcite, revealed numerous recent examples where genes had functionality similar to neighbors latest research. None of the authors in the most recent literature search were found using Google Scholar during the first literature search.

7.5 Limitations

The limitations of this study include areas related to GO terms that are not grouped, GO sets chosen for grouping, gene clusters chosen to study, and methods of annotating disease to genes.

7.5.1 GO terms not grouped

The ASCII art for the GO grouping excludes the GO terms placed into the "miscellaneous" section, which is a section for all GO terms that do not fit into sections defined by the researcher. GO terms that fall into the default group are not informative for determining whether two genes are similar since the default group is not specific since it is a catch-all for unspecified genes. It would be beneficial to examine all GO terms that were significant in the GOEAs of this thesis, but fell into "miscellaneous" section when grouped to perform the ASCII art visualization. More similarities may be identified if significant GO terms were moved into GO grouping sections.

7.5.2 GO sets chosen for grouping

Using ASCII art to visualize GO grouping currently uses the GO terms found to be significant in the GOEAs for each cluster rather than all GO terms annotated to a gene. It may be illuminating to use the full set of the GO terms associated with each gene during GO grouping visualization.

7.5.3 Gene clusters chosen

This study focused on examining protein clusters rather than disease gene clusters for genes with functionality similar to its neighbors. The protein-coding clusters and disease gene clusters overlap but overall are not the same clusters. Disease genes generally have more annotated GO terms compared to protein-coding genes that are not associated with many diseases. It may thus be interesting to examine genes' similarity when compared to neighbors in clusters of genes that are highly annotated for disease.

7.5.4 Disease annotation

Performing searches in NCBI Gene using disease names provided the list of diseases annotated to a gene. This is a free text search rather than a search for specific disease annotations, which can be performed using medical subject headings (MeSH) terms. The risk of a free text search is that some disease associations with a gene may represent a false positive.

For example, the gene involucrin appears to be associated with psoriasis while searching for "involucrin AND psoriasis" in PubMed but not while searching in NCBI Gene. Involucrin thus did not appear to be associated with

psoriasis in this study, which suggests that the disease annotation should be augmented with automated searches in PubMed. This is not a trivial task since using the gene symbols for genes such as acetylcholinesterase (ACHE), BCL2 associated agonist of cell death (BAD), and catalase (CAT) return many false positives; however using only the full name of a gene may not return all searches.

MeSH terms are lists of words grouped into trees and produced by the U.S. National Library of Medicine (NLM). MeSH terms are annotated onto research publications in the PubMed database and records in other NLM databases. One branch of MeSh terms is diseases, while others include anatomy, chemicals, and drugs.

Other methods to associate diseases and genes include ontologies annotated onto the gene, including disease ontologies, human phenotype ontologies, anatomy ontologies, etc. Haendel et al. described how ontologies from various sources may be combined for computational processes to support precision medicine by classifying patients for diagnosis, treatment, and translational research [81].

There are a many of disease and variant databases. The database Online Mendelian Inheritance in Man (OMIM) is an up-to-date extensive research resource of curated descriptions of the relationships between human genes and phenotypes [7]. ClinVar is a publicly available database linking human DNA variations and phenotypes through supporting evidence [112]. DisGeNET is a knowledge management tool suite that integrates data and scientific literature regarding how diseases are associated with human genes and variants [170]. The Leiden Open Variation Database (LOVD) is a freely available tool for

researching genomic variant and phenotype collection, and views in LOVD can be either patient centered or gene centered. Other variant databases include Sherloc [155], which uses guidelines from the American College of Medical Genetics and Genomics and the Association for Molecular Pathology (ACMG-AMP) [76]. Researchers have also found discrepancies between the classification of variants between the databases [208].

Choosing a database to use for annotations of disease to genes provides varying results depending on the choices. The method used in this study was to search NCBI's Gene database using disease names entered with free text. When checking sample gene-disease annotations by hand, the annotations were correct, but they were performed on a small sample, and thus false positives may exist. The data were accurate enough for this study, however, since the disease data were used to inform the scenery of the genome in the clusters. Future studies may benefit from using other databases.

7.5.5 Evolution and gene clusters

One hypothesized reason that genes are in clusters is so that they stay together over time, and another is that gene products in a cluster work together in a shared biological process [158]. If it is advantageous for genes to stay clustered together over time, it is likely that these clusters will stay together through speciation events. Studying orthologous genes that are retained in clusters across many species can thus lead to new insights.

Phylogenetic profiling is a bioinformatics method of identifying gene families that are apt to be lost or maintained across the tree of life in order to determine consequential biological connections, such as predicting genes

involved in the same pathway or biological process. In the past, phylogenetic profiling has more often been used on prokaryotes than eukaryotes, in part due to the significant size of most eukaryotic genomes compared to prokaryotes. Phylogenetic techniques customarily used on prokaryote genomes are too slow to be used on the much larger eukaryote genomes; however, new phylogenetic techniques in development that can be used on eukaryotic genomes, such as HogProf [141].

Phylogenetic profiling can only be used after determining which proteins are homologous to each other through different orthologs. Databases such as the orthologous matrix (OMA) help to to determine which genes are orthologous to others in other species [3]. The OMA database currently identifies orthologs across over 2,300 genomes across the tree of life [4].

Using orthologs in phylogenetic profiling can aid inferring which genes in a cluster share the same biological processes.

7.6 Future work

Because research on model organisms drives research on human genes and because the percentage of researchers working on model organisms has plummeted from over 90% of researchers to just over 40% [198], one area of future work is to add more information regarding the level of research coverage on model organism genes that are orthologous to human genes. Another area is improve and update gene functional semantic similarity methods.

7.6.1 Understudied genes in model organisms

Stoeger et al. found that research on model organisms drives research on human genes [198]. Wood et al. found that 20% of the proteins in well studied model organisms, such as yeast, lack thorough descriptions of their biological roles [216]. Many understudied genes in yeast are conserved from yeast to human, which suggests their importance in biological processes in human health and disease.

Understudied model organism genes represent blind spots in both applied health science and basic research. Wood et al. created a list of conserved but unstudied proteins in Schizosaccharomyces pombe (fission yeast) and classified them into GO groups using data from large-scale and comparative experiments.

Future work could add Wood's list of important understudied yeast genes that are conserved across yeast and human to my genome landscape visualization. This could be illuminating in choosing a set of understudied human genes to further study whose products are important in human health but might be more rapidly studied in a model organism. This could launch ideas for experiments on human cells.

7.6.2 Updating gene functional semantic similarity

There were significant, easy-to-find errors in the open-source gene functional semantic similarity method. It is thus worth using another perspective on the implementations of the various methods.

There are also improvements that can be made to the current gene functional semantic similarity methods, but such an investigation that would need to be scheduled as its own project. Members of the research community have expressed interest in further work on gene functional semantic similarity methods, as evidenced from comments made in GOA TOOLs GitHub issues.

7.6.3 The future

My goal is to facilitate groups of researchers to make discoveries that they would not have made otherwise. I look forward to collaborating with researchers from across the globe to advance human health. I am especially interested in research involving biological pathways and chemical reaction mechanisms to better visualize and examine the phenotypes affected by both the higher-level pathway view and the detailed chemical steps within the pathway.

Exploring biological pathways would build well on my experience with the GO because terms from all three branches (biological process (BP), molecular function (MF), and cellular component (CC)) are annotated to components in pathway databases such as Reactome [98]. For example, BP terms are annotated to most pathways in the Reactome database. Many pathways and reactions in Reactome are also annotated with CC terms to identify their location relative to cellular structures.

Pathways in the Reactome database contain physical entities such as molecular complexes, polymers, and DNA or RNA sequences which are inputs and outputs to chemical reactions that are annotated with MF terms. Molecular function is tied closely to organic chemistry and chemical ontologies

such as Chemical Entities of Biological Interest (ChEBI) [83]. I look forward to collaborating with researchers all over the Earth to make discoveries in biology and chemistry.

Acronyms

AG	Agglomerative Clustering
ALS	Amyotrophic Lateral Sclerosis
BCG	Biosynthetic Gene Cluster
CRP	C-reactive Protein
CXCL8	C-X-C motif chemokine Ligand
dme	fruit fly (Drosophila melanogaster)
GeneRIG	Gene Reference into Function
GO	Gene Ontology
GOEA	Gene Ontology Enrichment Analysis
GPCR	G-protein coupled receptors (non-olfactory)
GTR	NCBI's Genetic Testing Registry
GWA	Genome-Wide Association Studies
HGNC	HUGO Gene Nomenclature Committee
HGVS	Human Genome Variation Society
HLA	Human Leukocite Antigen
hsa	human (Homo sapiens)
HUGO	Human Genome Organisation
IDG	Illuminating the Druggable Genome
IL6	Interleukin 6
KS	Kolmogorov-Smirnov
lncRNA	Long noncoding RNAs
MeSH	Medical Subject Headings

MHC	Major Histocompatibility
mmu	mouse (Mus musculus)
NCBI	National Center for Biotechnology Information
NIH	National Institutes of Health
NIH-OCC	National Institute of Health's Open Citation Collection
NLM	U.S.
NN	Nearest Neighbor Clustering
NOS2	Nitric Oxide Synthase 2
oGPCR	olfactory G-protein coupled receptors
OMA	orthologous matrix
PCA	Principal component analysis
PPI	Protein-protein interaction
PTGS2	Prostaglandin-Endoperoxide Synthase 2
RCT	randomized controlled trials
RPI	Rensselaer Polytechnic Institute
SNF	Smith normal form
TDA	Topological data analysis
TDL	Target Development Level
TLR4	Toll Like Receptor 4
UCSC	University of Santa Cruz
VEGFA	Vascular Endothelial Growth Factor A

List of References

1. A. V. Aho, R. Sethi, and J. D. Ullman. "Compilers, Principles, Techniques". *Addison wesley* 7:8, 1986, p. 9.
2. A. Alexa and J. Rahnenfuhrer. "topGO: enrichment analysis for gene ontology". *R package version* 2:0, 2010.
3. A. M. Altenhoff, A. Schneider, G. H. Gonnet, and C. Dessimoz. "OMA 2011: orthology inference among 1000 complete genomes". *Nucleic Acids Research* 39:Database, 2010, pp. D289–D294. DOI: 10.1093/nar/gkq1238. pmcid: PMC3013747. URL: https://doi.org/10.1093%2Fnar%2Fgkq1238.
4. A. M. Altenhoff, C.-M. Train, K. J. Gilbert, I. Mediratta, T. Mendes de Farias, D. Moi, Y. Nevers, H.-S. Radoykova, V. Rossier, A. Warwick Vesztrocy, N. M. Glover, and C. Dessimoz. "OMA orthology in 2021: website overhaul, conserved isoforms, ancestral gene order and more". *Nucleic Acids Research* 49:D1, 2020, pp. D373–D379. DOI: 10.1093/nar/gkaa1007. pmcid: PMC7779010. URL: https://doi.org/10.1093%2Fnar%2Fgkaa1007.
5. G. Alterovitz, M. Xiang, M. Mohan, and M. F. Ramoni. "GO PaD: the gene ontology partition database". *Nucleic acids research* 35:suppl_1, 2006, pp. D322–D327.
6. J. S. Amberger, C. A. Bocchini, F. Schiettecatte, A. F. Scott, and A. Hamosh. "OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders". *Nucleic Acids Research* 43:D1, 2014, pp. D789–D798. DOI: 10.1093/nar/gku1205. pmcid: PMC4383985. URL: https://doi.org/10.1093%5C%2Fnar%5C%2Fgku1205.
7. J. S. Amberger and A. Hamosh. "Searching Online Mendelian Inheritance in Man (OMIM): A Knowledgebase of Human Genes and Genetic Phenotypes". *Current Protocols in Bioinformatics* 58:1, 2017. DOI: 10.1002/cpbi.27. pmcid: PMC5662200. URL: https://doi.org/10.1002%2Fcpbi.27.
8. J. van Arensbergen, V. D. FitzPatrick, M. de Haas, L. Pagie, J. Sluimer, H. J. Bussemaker, and B. van Steensel. "Genome-wide mapping of autonomous promoter activity in human cells". *Nature Biotechnology* 35:2, 2016, pp. 145–153. DOI: 10.1038/nbt.3754. pmcid: PMC5498152. URL: https://doi.org/10.1038%2Fnbt.3754.

9. A. W. Armstrong and C. Read. "Pathophysiology, Clinical Presentation, and Treatment of Psoriasis". *JAMA* 323:19, 2020, p. 1945. DOI: 10.1001/jama.2020.4006. PMID: 32427307. URL: https://doi.org/10.1001%2Fjama.2020.4006.

10. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. "Gene Ontology: tool for the unification of biology". *Nature Genetics* 25:1, 2000, pp. 25–29. DOI: 10.1038/75556. pmcid: PMC3037419. URL: https://doi.org/10.1038%5C%2F75556.

11. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. "Gene Ontology: tool for the unification of biology". *Nature genetics* 25:1, 2000, pp. 25–29.

12. K. Athukorala, D. Głowacka, G. Jacucci, A. Oulasvirta, and J. Vreeken. "Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks". *Journal of the Association for Information Science and Technology* 67:11, 2015, pp. 2635–2651. DOI: 10.1002/asi.23617. URL: https://doi.org/10.1002%5C%2Fasi.23617.

13. K. Athukorala, E. Hoggan, A. Lehtiö, T. Ruotsalo, and G. Jacucci. "Information-seeking behaviors of computer scientists: Challenges for electronic literature search tools". *Proceedings of the American Society for Information Science and Technology* 50:1, 2013, pp. 1–11. DOI: 10.1002/meet.14505001041. URL: https://doi.org/10.1002%5C%2Fmeet.14505001041.

14. A. A. Bazzini, F. del Viso, M. A. Moreno-Mateos, T. G. Johnstone, C. E. Vejnar, Y. Qin, J. Yao, M. K. Khokha, and A. J. Giraldez. "Codon identity regulates mRNA stability and translation efficiency during the maternal-to-zygotic transition". *The EMBO journal*, 2016, e201694699.

15. Y. Benjamini and Y. Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". *Journal of the Royal Statistical Society: Series B (Methodological)* 57:1, 1995, pp. 289–300. DOI: 10.1111/j.2517-6161.1995.tb02031.x. URL: https://doi.org/10.1111%2Fj.2517-6161.1995.tb02031.x.

16. D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. "GenBank". *Nucleic Acids Research* 41:D1, 2012, pp. D36–D42. DOI: 10.1093/nar/gks1195. pmcid: PMC3531190. URL: https://doi.org/10.1093%2Fnar%2Fgks1195.

17. S. D. Bentley, K. F. Chater, A. .-.-M. Cerdeño-Tárraga, G. L. Challis, N. R. Thomson, K. D. James, D. E. Harris, M. A. Quail, H. Kieser, D. Harper, A. Bateman, S. Brown, G. Chandra, C. W. Chen, M. Collins, A. Cronin, A. Fraser, A. Goble, J. Hidalgo, T. Hornsby, S. Howarth, C. .-.-H. Huang, T. Kieser, L. Larke, L. Murphy, K. Oliver, S. O'Neil, E. Rabbinowitsch, M. .-.-A. Rajandream, K. Rutherford, S. Rutter, K. Seeger, D. Saunders, S. Sharp, R. Squares, S. Squares, K. Taylor, T. Warren, A. Wietzorrek, J. Woodward, B. G. Barrell, J. Parkhill, and D. A. Hopwood. "Complete genome sequence of the model actinomycete Streptomyces coelicolor A3(2)". *Nature* 417:6885, 2002, pp. 141–147. DOI: 10.1038/417141a. PMID: 12000953. URL: https://doi.org/10.1038%2F417141a.

18. M. Bertolini, A. Ranjan, A. Thompson, P. I. Diaz, T. Sobue, K. Maas, and A. Dongari-Bagtzoglou. "Candida albicans induces mucosal bacterial dysbiosis that promotes invasive infection". *PLOS Pathogens* 15:4, 2019. Ed. by M. C. Noverr, e1007717. DOI: 10.1371/journal.ppat.1007717. URL: https://doi.org/10.1371%2Fjournal.ppat.1007717.

19. E. Björling and M. Uhlén. "Antibodypedia, a Portal for Sharing Antibody and Antigen Validation Data". *Molecular and Cellular Proteomics* 7:10, 2008, pp. 2028–2037. DOI: 10.1074/mcp.M800264-MCP200. PMID: 18667413. URL: https://doi.org/10.1074%5C%2Fmcp.m800264-mcp200.

20. M. V. Blagosklonny. "Aging, Stem Cells, and Mammalian Target of Rapamycin: A Prospect of Pharmacologic Rejuvenation of Aging Stem Cells". *Rejuvenation Research* 11:4, 2008, pp. 801–808. DOI: 10.1089/rej.2008.0722. PMID: 18729812. URL: https://doi.org/10.1089%2Frej.2008.0722.

21. J. A. Blake, J. T. Eppig, J. A. Kadin, J. E. Richardson, C. L. Smith, and C. J. Bult. "Mouse Genome Database (MGD)-2017: community knowledge resource for the laboratory mouse". *Nucleic acids research* 45:D1, 2017, pp. D723–D729.

22. M. Boeker, W. Vach, and E. Motschall. "Google Scholar as replacement for systematic literature searches: good relative recall and precision are not enough". *BMC Medical Research Methodology* 13:1, 2013. DOI: 10.1186/1471-2288-13-131. URL: https://doi.org/10.1186%5C%2F1471-2288-13-131.

23. E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock. "GO:: TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes". *Bioinformatics* 20:18, 2004, pp. 3710–3715.

24. B. Braschi, P. Denny, K. Gray, T. Jones, R. Seal, S. Tweedie, B. Yates, and E. Bruford. "Genenames.org: the HGNC and VGNC resources in 2019". *Nucleic Acids Research* 47:D1, 2018, pp. D786–D792. DOI: 10.1093/nar/gky930. pmcid: PMC6324057. URL: https://doi.org/10.1093%2Fnar%2Fgky930.

25. S. D. M. Brown and H. V. Lad. "The dark genome and pleiotropy: challenges for precision medicine". *Mammalian Genome* 30:7-8, 2019, pp. 212–216. DOI: 10.1007/s00335-019-09813-4. pmcid: PMC6759675. URL: https://doi.org/10.1007%5C%2Fs00335-019-09813-4.

26. R. A. Cairns, I. S. Harris, and T. W. Mak. "Regulation of cancer cell metabolism". *Nature Reviews Cancer* 11:2, 2011, pp. 85–95. DOI: 10.1038/nrc2981. PMID: 21258394. URL: https://doi.org/10.1038%2Fnrc2981.

27. E. Candi, R. Schmidt, and G. Melino. "The cornified envelope: a model of cell death in the skin". *Nature Reviews Molecular Cell Biology* 6:4, 2005, pp. 328–340. DOI: 10.1038/nrm1619. PMID: 15803139. URL: https://doi.org/10.1038%2Fnrm1619.

28. H. Caniza, A. E. Romero, S. Heron, H. Yang, A. Devoto, M. Frasca, M. Mesiti, G. Valentini, and A. Paccanaro. "GOssTo: a stand-alone application and a web tool for calculating semantic similarities on the Gene Ontology". *Bioinformatics* 30:15, 2014, pp. 2235–2236. DOI: 10.1093/bioinformatics/btu144. URL: https://doi.org/10.1093%5C%2Fbioinformatics%5C%2Fbtu144.

29. H. Caron, B. van Schaik, M. van der Mee, F. Baas, G. Riggins, P. van Sluis, M.-C. Hermus, R. van Asperen, K. Boon, P. A. Voûte, S. Heisterkamp, A. van Kampen, and R. Versteeg. "The Human Transcriptome Map: Clustering of Highly Expressed Genes in Chromosomal Domains". *Science* 291:5507, 2001, pp. 1289–1292. DOI: 10.1126/science.1056794. PMID: 11181992. URL: https://doi.org/10.1126%2Fscience.1056794.

30. W.-L. Charng, S. Yamamoto, and H. J. Bellen. "Shared mechanisms between Drosophila peripheral nervous system development and human neurodegenerative diseases". *Current Opinion in Neurobiology* 27, 2014, pp. 158–164. DOI: 10.1016/j.conb.2014.03.001. pmcid: PMC4122633. URL: https://doi.org/10.1016%5C%2Fj.conb.2014.03.001.

31. W. Chen, K. Mou, B. Xu, X. Ling, J. Cui, and P. Xu. "Capillary electrophoresis for screening of 20S proteasome inhibitors". *Analytical Biochemistry* 394:1, 2009, pp. 62–67. DOI: 10.1016/j.ab.2009.07.020. PMID: 19615965. URL: https://doi.org/10.1016%2Fj.ab.2009.07.020.

32. R.-l. Cheng, J. Feng, B.-X. Zhang, Y. Huang, J. Cheng, and C.-X. Zhang. "Transcriptome and gene expression analysis of an oleaginous diatom under different salinity conditions". *BioEnergy Research* 7:1, 2014, pp. 192–205.

33. M. C. Chibucos, C. J. Mungall, R. Balakrishnan, K. R. Christie, R. P. Huntley, O. White, J. A. Blake, S. E. Lewis, and M. Giglio. "Standardized description of scientific evidence using the Evidence Ontology (ECO)". *Database* 2014, 2014.

34. R. de Cid, E. Riveira-Munoz, P. L. J. M. Zeeuwen, J. Robarge, W. Liao, E. N. Dannhauser, E. Giardina, P. E. Stuart, R. Nair, C. Helms, G. Escaramıs, E. Ballana, G. Martın-Ezquerra, M. d. Heijer, M. Kamsteeg, I. Joosten, E. E. Eichler, C. Lázaro, R. M. Pujol, L. Armengol, G. Abecasis, J. T. Elder, G. Novelli, J. A. L. Armour, P.-Y. Kwok, A. Bowcock, J. Schalkwijk, and X. Estivill. "Deletion of the late cornified envelope LCE3B and LCE3C genes as a susceptibility factor for psoriasis". *Nature Genetics* 41:2, 2009, pp. 211–215. DOI: 10.1038/ng.313. pmcid: PMC3128734. URL: https://doi.org/10.1038%2Fng.313.

35. P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". *Bioinformatics* 25:11, 2009, pp. 1422–1423. DOI: 10.1093/bioinformatics/btp163. pmcid: PMC2682512. URL: https://doi.org/10.1093%2Fbioinformatics%2Fbtp163.

36. G. O. Consortium et al. "Expansion of the Gene Ontology knowledgebase and resources". *Nucleic acids research* 45:D1, 2017, pp. D331–D338.

37. M. Corrales, A. Rosado, R. Cortini, J. van Arensbergen, B. van Steensel, and G. J. Filion. "Clustering ofDrosophilahousekeeping promoters facilitates their expression". *Genome Research* 27:7, 2017, pp. 1153–1161. DOI: 10.1101/gr.211433.116. pmcid: PMC5495067. URL: https://doi.org/10.1101%2Fgr.211433.116.

38. F. M. Couto and M. J. Silva. "Disjunctive shared information between ontology concepts: application to Gene Ontology". *Journal of Biomedical Semantics* 2:1, 2011, p. 5. DOI: 10.1186/2041-1480-2-5. pmcid: PMC3200982. URL: https://doi.org/10.1186%5C%2F2041-1480-2-5.

39. F. M. Couto, M. J. Silva, and P. M. Coutinho. "Measuring semantic similarity between Gene Ontology terms". *Data and Knowledge Engineering* 61:1, 2007, pp. 137–152. DOI: 10.1016/j.datak.2006.05.003. URL: https://doi.org/10.1016%5C%2Fj.datak.2006.05.003.

40. P. Danecek, J. K. Bonfield, J. Liddle, J. Marshall, V. Ohan, M. O. Pollard, A. Whitwham, T. Keane, S. A. McCarthy, R. M. Davies, and H. Li. "Twelve years of SAMtools and BCFtools". *GigaScience* 10:2, 2021. giab008. ISSN: 2047-217X. DOI: 10.1093/gigascience/giab008. eprint: https://academic.oup.com/gigascience/article-pdf/10/2/giab008/36332246/giab008.pdf. URL: https://doi.org/10.1093/gigascience/giab008.

41. L. T. M. Dao, A. O. Galindo-Albarrán, J. A. Castro-Mondragon, C. Andrieu-Soler, A. Medina-Rivera, C. Souaid, G. Charbonnier, A. Griffon, L. Vanhille, T. Stephen, J. Alomairi, D. Martin, M. Torres, N. Fernandez, E. Soler, J. van Helden, D. Puthier, and S. Spicuglia. "Genome-wide characterization of mammalian promoters with distal enhancer functions". *Nature Genetics* 49:7, 2017, pp. 1073–1081. DOI: 10.1038/ng.3884. PMID: 28581502. URL: https://doi.org/10.1038%2Fng.3884.

42. N. S. De Silva and U. Klein. "Dynamics of B cells in germinal centres". *Nature Reviews Immunology* 15:3, 2015, pp. 137–148.

43. M. Demerec and P. E. Hartman. "Complex Loci in Microorganisms". *Annual Review of Microbiology* 13:1, 1959, pp. 377–406. DOI: 10.1146/annurev.mi.13.100159.002113. URL: https://doi.org/10.1146%5C%2Fannurev.mi.13.100159.002113.

44. D. Devos and A. Valencia. “Intrinsic errors in genome annotation”. *Trends in Genetics* 17:8, 2001, pp. 429–431. ISSN: 0168-9525. DOI: https://doi.org/10.1016/S0168-9525(01)02348-4. URL: http://www.sciencedirect.com/science/article/pii/S0168952501023484.

45. D. Devos and A. Valencia. “Practical limits of function prediction”. *Proteins: Structure, Function, and Bioinformatics* 41:1, 2000, pp. 98–107. DOI: 10.1002/1097-0134(20001001)41:1<98::AID-PROT120>3.0.CO;2-S. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1097-0134\%2820001001\%2941\%3A1\%3C98\%3A\%3AAID-PROT120\%3E3.0.CO\%3B2-S. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0134%5C%2820001001%5C%2941%5C%3A1%5C%3C98%5C%3A%5C%3AAID-PROT120%5C%3E3.0.CO%5C%3B2-S.

46. Y. Diao, R. Fang, B. Li, Z. Meng, J. Yu, Y. Qiu, K. C. Lin, H. Huang, T. Liu, R. J. Marina, I. Jung, Y. Shen, K.-L. Guan, and B. Ren. “A tiling-deletion-based genetic screen for cis-regulatory element identification in mammalian cells”. *Nature Methods* 14:6, 2017, pp. 629–635. DOI: 10.1038/nmeth.4264. pmcid: PMC5490986. URL: https://doi.org/10.1038%2Fnmeth.4264.

47. D. E. Dickel, I. Barozzi, Y. Zhu, Y. Fukuda-Yuzawa, M. Osterwalder, B. J. Mannion, D. May, C. H. Spurrell, I. Plajzer-Frick, C. S. Pickle, E. Lee, T. H. Garvin, M. Kato, J. A. Akiyama, V. Afzal, A. Y. Lee, D. U. Gorkin, B. Ren, E. M. Rubin, A. Visel, and L. A. Pennacchio. “Genome-wide compendium and functional assessment of in vivo heart enhancers”. *Nature Communications* 7:1, 2016. DOI: 10.1038/ncomms12923. pmcid: PMC5059478. URL: https://doi.org/10.1038%5C%2Fncomms12923.

48. S. Drăghici. *Statistics and data analysis for microarrays using R and bioconductor*. CRC Press, 2016.

49. S. Drăghici, P. Khatri, R. P. Martins, G. C. Ostermeier, and S. A. Krawetz. “Global functional profiling of gene expression”. *Genomics* 81:2, 2003, pp. 98–104. DOI: 10.1016/s0888-7543(02)00021-6. PMID: 12620386. URL: https://doi.org/10.1016%2Fs0888-7543%2802%2900021-6.

50. J. Duffin. *Stanley's dream : the Medical Expedition to Easter Island*. McGill-Queen’s University Press, Montreal Kingston London Chicago, 2019. ISBN: 9780773557109.

51. L. Duke. *College libraries and student culture : what we now know*. American Library Association, Chicago, 2011. ISBN: 9780838993576.

52. E. J. Duncan, M. P. Leask, and P. K. Dearden. "Genome Architecture Facilitates Phenotypic Plasticity in the Honeybee (Apis mellifera)". *Molecular Biology and Evolution* 37:7, 2020. Ed. by P. Wittkopp, pp. 1964–1978. DOI: 10.1093/molbev/msaa057. pmcid: PMC7306700. URL: https://doi.org/10.1093%2Fmolbev%2Fmsaa057.

53. M. Edidin. "MHC antigens and non-immune functions". *Immunology Today* 4:10, 1983, pp. 269–270. DOI: 10.1016/0167-5699(83)90129-9. PMID: 25290508. URL: https://doi.org/10.1016%2F0167-5699%2883%2990129-9.

54. P.-H. D. Edqvist, L. Fagerberg, B. M. Hallström, A. Danielsson, K. Edlund, M. Uhlén, and F. Pontén. "Expression of Human Skin-Specific Genes Defined by Transcriptomics and Antibody-Based Profiling". *Journal of Histochemistry & Cytochemistry* 63:2, 2014, pp. 129–141. DOI: 10.1369/0022155414562646. pmcid: PMC4305515. URL: https://doi.org/10.1369%2F0022155414562646.

55. A. M. Edwards, R. Isserlin, G. D. Bader, S. V. Frye, T. M. Willson, and F. H. Yu. "Too many roads not taken". *Nature* 470:7333, 2011, pp. 163–165. DOI: 10.1038/470163a. PMID: 21307913. URL: https://doi.org/10.1038%5C%2F470163a.

56. M. E. Falagas, E. I. Pitsouni, G. A. Malietzis, and G. Pappas. "Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses". *The FASEB Journal* 22:2, 2008, pp. 338–342. DOI: 10.1096/fj.07-9492lsf. URL: https://doi.org/10.1096%5C%2Ffj.07-9492lsf.

57. S. Falcon and R. Gentleman. "Using GOstats to test gene lists for GO term association". *Bioinformatics* 23:2, 2006, pp. 257–258.

58. C. Finan, A. Gaulton, F. A. Kruger, R. T. Lumbers, T. Shah, J. Engmann, L. Galver, R. Kelley, A. Karlsson, R. Santos, J. P. Overington, A. D. Hingorani, and J. P. Casas. "The druggable genome and support for target identification and validation in drug development". *Science Translational Medicine* 9:383, 2017, eaag1166. DOI: 10.1126/scitranslmed.aag1166. pmcid: PMC6321762. URL: https://doi.org/10.1126%5C%2Fscitranslmed.aag1166.

59. R. D. Finn, T. K. Attwood, P. C. Babbitt, A. Bateman, P. Bork, A. J. Bridge, H.-Y. Chang, Z. Dosztányi, S. El-Gebali, M. Fraser, J. Gough, D. Haft, G. L. Holliday, H. Huang, X. Huang, I. Letunic, R. Lopez, S. Lu, A. Marchler-Bauer, H. Mi, J. Mistry, D. A. Natale, M. Necci, G. Nuka, C. A. Orengo, Y. Park, S. Pesseat, D. Piovesan, S. C. Potter, N. D. Rawlings, N. Redaschi, L. Richardson, C. Rivoire, A. Sangrador-Vegas, C. Sigrist, I. Sillitoe, B. Smithers, S. Squizzato, G. Sutton, N. Thanki, P. D. Thomas, S. Tosatto, C. H. Wu, I. Xenarios, L.-S. Yeh, S.-Y. Young, and A. L. Mitchell. "InterPro in 2017—beyond protein family and domain annotations". *Nucleic Acids Research* 45:D1, 2016, pp. D190–D199. DOI: 10.1093/nar/gkw1107. pmcid: PMC5210578. URL: https://doi.org/10.1093%5C%2Fnar%5C%2Fgkw1107.

60. N. Fiorini, K. Canese, R. Bryzgunov, I. Radetska, A. Gindulyte, M. Latterner, V. Miller, M. Osipov, M. Kholodov, G. Starchenko, E. Kireev, and Z. Lu. "PubMed Labs: an experimental system for improving biomedical literature search". *Database* 2018, 2018. DOI: 10.1093/database/bay094. URL: https://doi.org/10.1093%5C%2Fdatabase%5C%2Fbay094.

61. N. Fiorini, K. Canese, G. Starchenko, E. Kireev, W. Kim, V. Miller, M. Osipov, M. Kholodov, R. Ismagilov, S. Mohan, J. Ostell, and Z. Lu. "Best Match: New relevance search for PubMed". *PLOS Biology* 16:8, 2018, e2005343. DOI: 10.1371/journal.pbio.2005343. URL: https://doi.org/10.1371%5C%2Fjournal.pbio.2005343.

62. N. Fiorini, D. J. Lipman, and Z. Lu. "Towards PubMed 2.0". *eLife* 6, 2017. DOI: 10.7554/elife.28801. URL: https://doi.org/10.7554%5C%2Felife.28801.

63. M. A. Fischbach, C. T. Walsh, and J. Clardy. "The evolution of gene collectives: How natural selection drives chemical innovation". *Proceedings of the National Academy of Sciences* 105:12, 2008, pp. 4601–4608. DOI: 10.1073/pnas.0709132105. pmcid: PMC2290807. URL: https://doi.org/10.1073%2Fpnas.0709132105.

64. R. Fredriksson, M. C. Lagerström, L.-G. Lundin, and H. B. Schiöth. "The G-Protein-Coupled Receptors in the Human Genome Form Five Main Families. Phylogenetic Analysis, Paralogon Groups, and Fingerprints". *Molecular Pharmacology* 63:6, 2003, pp. 1256–1272. DOI: 10.1124/mol.63.6.1256. PMID: 12761335. URL: https://doi.org/10.1124%2Fmol.63.6.1256.

65. M. Frey. "Analysis of a Chemical Plant Defense Mechanism in Grasses". *Science* 277:5326, 1997, pp. 696–699. DOI: 10.1126/science.277.5326.696. PMID: 9235894. URL: https://doi.org/10.1126%2Fscience.277.5326.696.

66. A. D. Furlan, V. Pennick, C. Bombardier, and M. van Tulder. "2009 Updated Method Guidelines for Systematic Reviews in the Cochrane Back Review Group". *Spine* 34:18, 2009, pp. 1929–1941. DOI: 10.1097/BRS.0b013e3181b1c99f. PMID: 19680101. URL: https://doi.org/10.1097%5C%2Fbrs.0b013e3181b1c99f.

67. P. Gaudet and C. Dessimoz. "Gene ontology: pitfalls, biases, and remedies". *The Gene Ontology Handbook*, 2017, pp. 189–205.

68. J.-F. Gehanno, L. Rollin, and S. Darmoni. "Is the coverage of google scholar enough to be used alone for systematic reviews". *BMC Medical Informatics and Decision Making* 13:1, 2013. DOI: 10.1186/1472-6947-13-7. URL: https://doi.org/10.1186%5C%2F1472-6947-13-7.

69. "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls". *Nature* 447:7145, 2007, pp. 661–678. DOI: 10.1038/nature05911. pmcid: PMC2719288. URL: https://doi.org/10.1038%2Fnature05911.

70. H. Georgas. "Google vs. the Library: Student Preferences and Perceptions When Doing Research Using Google and a Federated Search Tool". *portal: Libraries and the Academy* 13:2, 2013, pp. 165–185. DOI: 10.1353/pla.2013.0011. URL: https://doi.org/10.1353%5C%2Fpla.2013.0011.

71. A. T. Ghanbarian and L. D. Hurst. "Neighboring Genes Show Correlated Evolution in Gene Expression". *Molecular Biology and Evolution* 32:7, 2015, pp. 1748–1766. DOI: 10.1093/molbev/msv053. pmcid: PMC4476153. URL: https://doi.org/10.1093%2Fmolbev%2Fmsv053.

72. D. Giustini and M. N. K. Boulos. "Google Scholar is not enough to be used alone for systematic reviews". *Online Journal of Public Health Informatics* 5:2, 2013. DOI: 10.5210/ojphi.v5i2.4623. URL: https://doi.org/10.5210%5C%2Fojphi.v5i2.4623.

73. E. Gjoneska, A. R. Pfenning, H. Mathys, G. Quon, A. Kundaje, L.-H. Tsai, and M. Kellis. "Conserved epigenomic signals in mice and humans reveal immune basis of Alzheimer/'s disease". *Nature* 518:7539, 2015, pp. 365–369.

74. M. E. Glickman, S. R. Rao, and M. R. Schultz. "False discovery rate control is a recommended alternative to Bonferroni-type adjustments in health studies". *Journal of clinical epidemiology* 67:8, 2014, pp. 850–857.

75. J. J. Goeman and A. Solari. "Multiple hypothesis testing in genomics". *Statistics in medicine* 33:11, 2014, pp. 1946–1978.

76. R. C. Green, J. S. Berg, W. W. Grody, S. S. Kalia, B. R. Korf, C. L. Martin, A. L. McGuire, R. L. Nussbaum, J. M. O'Daniel, K. E. Ormond, H. L. Rehm, M. S. Watson, M. S. Williams, and L. G. Biesecker. "ACMG recommendations for reporting of incidental findings in clinical exome and genome sequencing". *Genetics in Medicine* 15:7, 2013, pp. 565–574. DOI: 10.1038/gim.2013.73. pmcid: PMC3727274. URL: https://doi.org/10.1038%2Fgim.2013.73.

77. G. S. C. GSC. *MIBiG 2.0 Nov 20, 2020*. https://mibig.secondarymetabolites.org/stats. [Online; accessed 2020-11-20]. 2020.

78. M. Gusenbauer. "Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases". *Scientometrics* 118:1, 2018, pp. 177–214. DOI: 10.1007/s11192-018-2958-5. URL: https://doi.org/10.1007%5C%2Fs11192-018-2958-5.

79. M. Gusenbauer and N. R. Haddaway. "What every Researcher should know about Searching – Clarified Concepts, Search Advice, and an Agenda to improve Finding in Academia". *Research Synthesis Methods*, 2020. DOI: 10.1002/jrsm.1457. PMID: 33031639. URL: https://doi.org/10.1002%2Fjrsm.1457.

80. M. Gusenbauer and N. R. Haddaway. "Which Academic Search Systems are Suitable for Systematic Reviews or Meta-Analyses? Evaluating Retrieval Qualities of Google Scholar, PubMed and 26 other Resources". *Research Synthesis Methods*, 2019. DOI: 10.1002/jrsm.1378. URL: https://doi.org/10.1002%5C%2Fjrsm.1378.

81. M. A. Haendel, C. G. Chute, and P. N. Robinson. "Classification, Ontology, and Precision Medicine". *New England Journal of Medicine* 379:15, 2018. Ed. by E. G. Phimister, pp. 1452–1462. DOI: 10.1056/NEJMra1615014. pmcid: PMC6503847. URL: https://doi.org/10.1056%2Fnejmra1615014.

82. J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigo, and T. J. Hubbard. "GENCODE: The reference human genome annotation for The ENCODE Project". *Genome Research* 22:9, 2012, pp. 1760–1774. DOI: 10.1101/gr.135350.111. pmcid: PMC3431492. URL: https://doi.org/10.1101%2Fgr.135350.111.

83. J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, and C. Steinbeck. "ChEBI in 2016: Improved services and an expanding collection of metabolites". *Nucleic Acids Research* 44:D1, 2015, pp. D1214–D1219. DOI: 10.1093/nar/gkv1031. pmcid: PMC4702775. URL: https://doi.org/10.1093%2Fnar%2Fgkv1031.

84. B. M. Hemminger, D. Lu, K. T. L. Vaughan, and S. J. Adams. "Information seeking behavior of academic scientists". *Journal of the American Society for Information Science and Technology* 58:14, 2007, pp. 2205–2225. DOI: 10.1002/asi.20686. URL: https://doi.org/10.1002%5C%2Fasi.20686.

85. L. K. HENDERSON, J. C. CRAIG, N. S. WILLIS, D. TOVEY, and A. C. WEBSTER. "How to write a Cochrane systematic review". *Nephrology* 15:6, 2010, pp. 617–624. DOI: 10.1111/j.1440-1797.2010.01380.x. PMID: 20883282. URL: https://doi.org/10.1111%5C%2Fj.1440-1797.2010.01380.x.

86. J. Henry. "Update on the epidermal differentiation complex". *Frontiers in Bioscience* 17:1, 2012, p. 1517. DOI: 10.2741/4001. PMID: 22201818. URL: https://doi.org/10.2741%2F4001.

87. B. Hjørland. "Classical databases and knowledge organization: A case for boolean retrieval and human decision-making during searches". *Journal of the Association for Information Science and Technology* 66:8, 2014, pp. 1559–1575. DOI: 10.1002/asi.23250. URL: https://doi.org/10.1002%5C%2Fasi.23250.

88. D. A. Hopwood. "Forty years of genetics with Streptomyces: from in vivo through in vitro to in silico". *Microbiology* 145:9, 1999, pp. 2183–2202. DOI: 10.1099/00221287-145-9-2183. PMID: 10517572. URL: https://doi.org/10.1099%2F00221287-145-9-2183.

89. D. W. Huang, B. T. Sherman, and R. A. Lempicki. "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists". *Nucleic acids research* 37:1, 2008, pp. 1–13.

90. D. W. Huang, B. T. Sherman, and R. A. Lempicki. "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources". *Nature protocols* 4:1, 2008, p. 44.

91. T. Hubbard. "The Ensembl genome database project". *Nucleic Acids Research* 30:1, 2002, pp. 38–41. DOI: 10.1093/nar/30.1.38. pmcid: PMC99161. URL: https://doi.org/10.1093%2Fnar%2F30.1.38.

92. B. I. Hutchins, K. L. Baker, M. T. Davis, M. A. Diwersy, E. Haque, R. M. Harriman, T. A. Hoppe, S. A. Leicht, P. Meyer, and G. M. Santangelo. "The NIH Open Citation Collection: A public access, broad coverage resource". *PLOS Biology* 17:10, 2019, e3000385. DOI: 10.1371/journal.pbio.3000385. URL: https://doi.org/10.1371%5C%2Fjournal.pbio.3000385.

93. B. I. Hutchins, X. Yuan, J. M. Anderson, and G. M. Santangelo. "Relative Citation Ratio (RCR): A New Metric That Uses Citation Rates to Measure Influence at the Article Level". *PLOS Biology* 14:9, 2016. Ed. by D. L. Vaux, e1002541. DOI: 10.1371/journal.pbio.1002541. URL: https://doi.org/10.1371%5C%2Fjournal.pbio.1002541.

94. *I have noticed an error in a court opinion you are providing. What I can do to help fix it?* Accessed: 2021-05-05. 2020.

95. "Initial sequencing and analysis of the human genome". *Nature* 409:6822, 2001, pp. 860–921. DOI: 10.1038/35057062. PMID: 11237011. URL: https://doi.org/10.1038%2F35057062.

96. F. Jacob, D. Perrin, C. Sánchez, and J. Monod. "L'opéron : groupe de gènes à expression coordonnée par un opérateur [C. R. Acad. Sci. Paris 250 (1960) 1727–1729]". *Comptes Rendus Biologies* 328:6, 2005, pp. 514–520. DOI: 10.1016/j.crvi.2005.04.005. PMID: 15999435. URL: https://doi.org/10.1016%2Fj.crvi.2005.04.005.

97. H. R. Jamali and S. Asadi. "Google and the scholar: the role of Google in scientists' information-seeking behaviour". *Online Information Review* 34:2, 2010, pp. 282–294. DOI: 10.1108/14684521011036990. URL: https://doi.org/10.1108%5C%2F14684521011036990.

98. B. Jassal, L. Matthews, G. Viteri, C. Gong, P. Lorente, A. Fabregat, K. Sidiropoulos, J. Cook, M. Gillespie, R. Haw, F. Loney, B. May, M. Milacic, K. Rothfels, C. Sevilla, V. Shamovsky, S. Shorser, T. Varusai, J. Weiser, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio. "The reactome pathway knowledgebase". *Nucleic Acids Research*, 2019. DOI: 10.1093/nar/gkz1031. pmcid: PMC7145712. URL: https://doi.org/10.1093%2Fnar%2Fgkz1031.

99. J. J. Jiang and D. W. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". *arXiv preprint cmp-lg/9709008*, 1997.

100. S. C. Johnson. *Lint, a C program checker*. Bell Telephone Laboratories Murray Hill, 1977.

101. S. Karimi, S. Pohl, F. Scholer, L. Cavedon, and J. Zobel. "Boolean versus ranked querying for biomedical systematic reviews". *BMC Medical Informatics and Decision Making* 10:1, 2010. DOI: 10.1186/1472-6947-10-58. URL: https://doi.org/10.1186%5C%2F1472-6947-10-58.

102. B. A. Kidd, L. A. Peters, E. E. Schadt, and J. T. Dudley. "Unifying immunology with informatics and multiscale biology". *Nature Immunology* 15:2, 2014, pp. 118–127. DOI: 10.1038/ni.2787. pmcid: PMC4345400. URL: https://doi.org/10.1038%2Fni.2787.

103. D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg. "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions". *Genome Biology* 14:4, 2013, R36. DOI: 10.1186/gb-2013-14-4-r36. pmcid: PMC4053844. URL: https://doi.org/10.1186%2Fgb-2013-14-4-r36.

104. J. Kind, L. Pagie, H. Ortabozkoyun, S. Boyle, S. de Vries, H. Janssen, M. Amendola, L. Nolen, W. Bickmore, and B. van Steensel. "Single-Cell Dynamics of Genome-Nuclear Lamina Interactions". *Cell* 153:1, 2013, pp. 178–192. DOI: 10.1016/j.cell.2013.02.028. PMID: 23523135. URL: https://doi.org/10.1016%2Fj.cell.2013.02.028.

105. D. V. Klopfenstein and W. Dampier. "Commentary to Gusenbauer and Haddaway 2020: Evaluating retrieval qualities of Google Scholar and PubMed". *Research Synthesis Methods*, 2020. DOI: 10.1002/jrsm.1456. PMID: 33031632. URL: https://doi.org/10.1002%2Fjrsm.1456.

106. D. V. Klopfenstein, L. Zhang, B. S. Pedersen, F. Ramírez, A. W. Vesztrocy, A. Naldi, C. J. Mungall, J. M. Yunes, O. Botvinnik, M. Weigel, W. Dampier, C. Dessimoz, P. Flick, and H. Tang. "GOATOOLS: A Python library for Gene Ontology analyses". *Scientific Reports* 8:1, 2018.

DOI: 10.1038/s41598-018-28948-z. URL: https://doi.org/10.1038%5C%2Fs41598-018-28948-z.

107. S. T. Kosak. "Gene Order and Dynamic Domains". *Science* 306:5696, 2004, pp. 644–647. DOI: 10.1126/science.1103864. PMID: 15499009. URL: https://doi.org/10.1126%2Fscience.1103864.

108. G. Koscielny, G. Yaikhom, V. Iyer, T. F. Meehan, H. Morgan, J. Atienza-Herrero, A. Blake, C.-K. Chen, R. Easty, A. D. Fenza, T. Fiegel, M. Grifiths, A. Horne, N. A. Karp, N. Kurbatova, J. C. Mason, P. Matthews, D. J. Oakley, A. Qazi, J. Regnart, A. Retha, L. A. Santos, D. J. Sneddon, J. Warren, H. Westerberg, R. J. Wilson, D. G. Melvin, D. Smedley, S. D. M. Brown, P. Flicek, W. C. Skarnes, A.-M. Mallon, and H. Parkinson. "The International Mouse Phenotyping Consortium Web Portal, a unified point of access for knockout mice and related phenotyping data". *Nucleic Acids Research* 42:D1, 2013, pp. D802–D809. DOI: 10.1093/nar/gkt977. pmcid: PMC3964955. URL: https://doi.org/10.1093%5C%2Fnar%5C%2Fgkt977.

109. A. Krogh and J. Wang. "MHC region and its related disease study". *University of Copenhagen*, 2015. [Online; accessed 2020-08-22].

110. W. de Laat and F. Grosveld. *Chromosome Research* 11:5, 2003, pp. 447–459. DOI: 10.1023/a:1024922626726. PMID: 12971721. URL: https://doi.org/10.1023%2Fa%3A1024922626726.

111. J. Labbadia and R. I. Morimoto. "The Biology of Proteostasis in Aging and Disease". *Annual Review of Biochemistry* 84:1, 2015, pp. 435–464. DOI: 10.1146/annurev-biochem-060614-033955. pmcid: PMC4539002. URL: https://doi.org/10.1146%2Fannurev-biochem-060614-033955.

112. M. J. Landrum, S. Chitipiralla, G. R. Brown, C. Chen, B. Gu, J. Hart, D. Hoffman, W. Jang, K. Kaur, C. Liu, V. Lyoshin, Z. Maddipatla, R. Maiti, J. Mitchell, N. O'Leary, G. R. Riley, W. Shi, G. Zhou, V. Schneider, D. Maglott, J. B. Holmes, and B. L. Kattman. "ClinVar: improvements to accessing data". *Nucleic Acids Research* 48:D1, 2019, pp. D835–D844. DOI: 10.1093/nar/gkz972. pmcid: PMC6943040. URL: https://doi.org/10.1093%2Fnar%2Fgkz972.

113. B. Lenhard, A. Sandelin, and P. Carninci. "Metazoan promoters: emerging characteristics and insights into transcriptional regulation". *Nature Reviews Genetics* 13:4, 2012, pp. 233–245. DOI: 10.1038/nrg3163. PMID: 22392219. URL: https://doi.org/10.1038%2Fnrg3163.

114. M. J. Lercher. "Coexpression of Neighboring Genes in Caenorhabditis Elegans Is Mostly Due to Operons and Duplicate Genes". *Genome Research* 13:2, 2003, pp. 238–243. DOI: 10.1101/gr.553803. pmcid: PMC420373. URL: https://doi.org/10.1101%2Fgr.553803.

115. E. B. Lewis. "The Phenomenon of Position Effect". In: *Advances in Genetics*. Elsevier, 1950, pp. 73–115. DOI: 10.1016/s0065-2660(08)60083-8. PMID: 15425389. URL: https://doi.org/10.1016%2Fs0065-2660%2808%2960083-8.

116. M. Lewis. "The lac repressor". *Comptes Rendus Biologies* 328:6, 2005, pp. 521–548. DOI: 10.1016/j.crvi.2005.04.004. PMID: 15950160. URL: https://doi.org/10.1016%2Fj.crvi.2005.04.004.

117. G. Li, Y. Zhao, Z. Liu, C. Gao, F. Yan, B. Liu, and J. Feng. "De novo assembly and characterization of the spleen transcriptome of common carp (Cyprinus carpio) using Illumina paired-end sequencing". *Fish & shellfish immunology* 44:2, 2015, pp. 420–429.

118. M. Li, Y. Wu, G. Chen, Y. Yang, D. Zhou, Z. Zhang, D. Zhang, Y. Chen, Z. Lu, L. He, J. Zheng, and Y. Liu. "Deletion of the Late Cornified Envelope Genes LCE3C and LCE3B Is Associated with Psoriasis in a Chinese Population". *Journal of Investigative Dermatology* 131:8, 2011, pp. 1639–1643. DOI: 10.1038/jid.2011.86. PMID: 21509048. URL: https://doi.org/10.1038%2Fjid.2011.86.

119. Z. Li, X. Jiao, G. D. Sante, A. Ertel, M. C. Casimiro, M. Wang, S. Katiyar, X. Ju, D. V. Klopfenstein, A. Tozeren, W. Dampier, I. Chepelev, A. Jeltsch, and R. G. Pestell. "Cyclin D1 integrates G9a-mediated histone methylation". *Oncogene* 38:22, 2019, pp. 4232–4249. DOI: 10.1038/s41388-019-0723-8. pmcid: PMC6542714. URL: https://doi.org/10.1038%2Fs41388-019-0723-8.

120. Y.-C. Liang, P. Wu, G.-W. Lin, C.-K. Chen, C.-Y. Yeh, S. Tsai, J. Yan, T.-X. Jiang, Y.-C. Lai, D. Huang, M. Cai, R. Choi, R. B. Widelitz, W. Lu, and C.-M. Chuong. "Folding Keratin Gene Clusters during Skin Regional Specification". *Developmental Cell* 53:5, 2020, 561–576.e9. DOI: 10.1016/j.devcel.2020.05.007. pmcid: PMC7386462. URL: https://doi.org/10.1016%2Fj.devcel.2020.05.007.

121. D. Lin et al. "An information-theoretic definition of similarity." In: *Icml*. Vol. 98. 1998. 1998, pp. 296–304.

122. D. Lin. "An information-theoretic definition of similarity." In: *Icml*. Vol. 98. 1998. Citeseer. 1998, pp. 296–304.

123. Y. Liu, C. Helms, W. Liao, L. C. Zaba, S. Duan, J. Gardner, C. Wise, A. Miner, M. J. Malloy, C. R. Pullinger, J. P. Kane, S. Saccone, J. Worthington, I. Bruce, P. Kwok, A. Menter, J. Krueger, A. Barton, N. L. Saccone, and A. M. Bowcock. "A Genome-Wide Association Study of Psoriasis and Psoriatic Arthritis Identifies New Disease Loci". *PLoS Genetics* 4:4, 2008. Ed. by S. M. Leal, e1000041. DOI: 10.1371/journal.pgen.1000041. pmcid: PMC2274885. URL: https://doi.org/10.1371%2Fjournal.pgen.1000041.

124. S. Lohr. *Google Schools Its Algorithm*. lhttps://www.nytimes.com/2011/03/06/weekinreview/06lohr.html. Accessed: 2020-01-13. 2011.

125. E. D. López-Cózar, N. Robinson-García, and D. Torres-Salinas. "The Google scholar experiment: How to index false papers and manipulate bibliometric indicators". *Journal of the Association for Information Science and Technology* 65:3, 2013, pp. 446–454. DOI: 10.1002/asi.23056. URL: https://doi.org/10.1002%5C%2Fasi.23056.

126. P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation". *Bioinformatics* 19:10, 2003, pp. 1275–1283.

127. C. M. *The New PubMed is Here*. (Accessed 2019-12-05). 2019. URL: https://www.nlm.nih.gov/pubs/techbull/nd19/nd19_pubmed_new.html (visited on 11/18/2019).

128. I. C. MacLennan. "Germinal centers". *Annual review of immunology* 12:1, 1994, pp. 117–139.

129. F. Malpartida and D. A. Hopwood. "Molecular cloning of the whole biosynthetic pathway of a Streptomyces antibiotic and its expression in a heterologous host". *Nature* 309:5967, 1984, pp. 462–464. DOI: 10.1038/309462a0. PMID: 6328317. URL: https://doi.org/10.1038%2F309462a0.

130. M. Marcet-Houben and T. Gabaldón. "Evolutionary and functional patterns of shared gene neighbourhood in fungi". *Nature Microbiology* 4:12, 2019, pp. 2383–2392. DOI: 10.1038/s41564-019-0552-0. PMID: 31527797. URL: https://doi.org/10.1038%2Fs41564-019-0552-0.

131. R. R. Martel, J. Klicius, and S. Galet. "Inhibition of the immune response by rapamycin, a new antifungal antibiotic". *Canadian Journal of Physiology and Pharmacology* 55:1, 1977, pp. 48–51. DOI: 10.1139/y77-007. PMID: 843990. URL: https://doi.org/10.1139%2Fy77-007.

132. P. Mayr and A.-K. Walter. "An exploratory study of Google Scholar". *Online information review* 31:6, 2007, pp. 814–830.

133. G. K. Mazandu, E. R. Chimusa, M. Mbiyavanga, and N. J. Mulder. "A-DaGO-Fun: an adaptable Gene Ontology semantic similarity-based functional analysis tool". *Bioinformatics* 32:3, 2015, pp. 477–479. DOI: 10.1093/bioinformatics/btv590. pmcid: PMC5006308. URL: https://doi.org/10.1093%2Fbioinformatics%2Fbtv590.

134. G. K. Mazandu, E. R. Chimusa, and N. J. Mulder. "Gene Ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery". *Briefings in Bioinformatics*, 2016, bbw067. DOI: 10.1093/bib/bbw067. PMID: 27473066. URL: https://doi.org/10.1093%2Fbib%2Fbbw067.

135. G. K. Mazandu and N. J. Mulder. "Information Content-Based Gene Ontology Functional Similarity Measures: Which One to Use for a Given Biological Data Type?" *PLoS ONE* 9:12, 2014. Ed. by C. Gibas, e113859. DOI: 10.1371/journal.pone.0113859. pmcid: PMC4256219. URL: https://doi.org/10.1371%2Fjournal.pone.0113859.

136. J. H. McDonald. *Handbook of biological statistics*. Vol. 2. Sparky House Publishing Baltimore, MD, 2009.

137. L. McKeever, V. Nguyen, S. J. Peterson, S. Gomez-Perez, and C. Braunschweig. "Demystifying the Search Button". *Journal of Parenteral and Enteral Nutrition* 39:6, 2015, pp. 622–635. DOI: 10.1177/0148607115593791. URL: https://doi.org/10.1177%5C%2F0148607115593791.

138. M. H. Medema et al. "Minimum Information about a Biosynthetic Gene cluster". *Nature Chemical Biology* 11:9, 2015, pp. 625–631. DOI: 10.1038/nchembio.1890. pmcid: PMC5714517. URL: https://doi.org/10.1038%2Fnchembio.1890.

139. P. Michalak. "Coexpression, coregulation, and cofunctionality of neighboring genes in eukaryotic genomes". *Genomics* 91:3, 2008, pp. 243–248. DOI: 10.1016/j.ygeno.2007.11.002. PMID: 18082363. URL: https://doi.org/10.1016%2Fj.ygeno.2007.11.002.

140. J. A. Mitchell, A. R. Aronson, J. G. Mork, L. C. Folk, S. M. Humphrey, and J. M. Ward. *Gene Indexing: Characterization and Analysis of NLM's GeneRIFs*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1480312/. [Online; accessed 2020-07-29]. 2020.

141. D. Moi, L. Kilchoer, P. S. Aguilar, and C. Dessimoz. "Scalable phylogenetic profiling using MinHash uncovers likely eukaryotic sexual reproduction genes". *PLOS Computational Biology* 16:7, 2020. Ed. by C. A. Ouzounis, e1007553. DOI: 10.1371/journal.pcbi.1007553. pmcid: PMC7423146. URL: https://doi.org/10.1371%2Fjournal.pcbi.1007553.

142. S. T. Mugford, X. Qi, S. Bakht, L. Hill, E. Wegel, R. K. Hughes, K. Papadopoulou, R. Melton, M. Philo, F. Sainsbury, G. P. Lomonossoff, A. D. Roy, R. J. M. Goss, and A. Osbourn. "A Serine Carboxypeptidase-Like Acyltransferase Is Required for Synthesis of Antimicrobial Compounds and Disease Resistance in Oats". *The Plant Cell* 21:8, 2009, pp. 2473–2484. DOI: 10.1105/tpc.109.065870. pmcid: PMC2751944. URL: https://doi.org/10.1105%2Ftpc.109.065870.

143. V. Muñoz-Fuentes, P. Cacheiro, T. F. Meehan, J. A. Aguilar-Pimentel, S. D. M. Brown, A. M. Flenniken, P. Flicek, A. Galli, H. H. Mashhadi, M. H. de Angelis, J. K. Kim, K. C. K. Lloyd, C. McKerlie, H. Morgan, S. A. Murray, L. M. J. Nutter, P. T. Reilly, J. R. Seavitt, J. K. Seong, M. Simon, H. Wardle-Jones, A.-M. Mallon, D. Smedley, and H. E. Parkinson. "The International Mouse Phenotyping Consortium (IMPC): a functional catalogue of the mammalian genome that informs conservation". *Conservation Genetics* 19:4, 2018, pp. 995–1005. DOI: 10.1007/s10592-018-1072-9. pmcid: PMC6061128. URL: https://doi.org/10.1007%5C%2Fs10592-018-1072-9.

144. F. Murtagh. "A Survey of Recent Advances in Hierarchical Clustering Algorithms". *The Computer Journal* 26:4, 1983, pp. 354–359. DOI: 10.1093/comjnl/26.4.354. URL: https://doi.org/10.1093%2Fcomjnl%2F26.4.354.

145. G. Naik. "The quiet rise of the NIH's hot new metric". *Nature* 539:7628, 2016, pp. 150–150. DOI: 10.1038/539150a. URL: https://doi.org/10.1038%5C%2F539150a.

146. A. NCBI. *Entrez Programming Utilities Help [Internet]*. https://www.ncbi.nlm.nih.gov/books/NBK25501/. [Online; accessed 2021-06-02]. 2010.

147. M. R. Nelson, H. Tipney, J. L. Painter, J. Shen, P. Nicoletti, Y. Shen, A. Floratos, P. C. Sham, M. J. Li, J. Wang, L. R. Cardon, J. C. Whittaker, and P. Sanseau. "The support of human genetic evidence for approved drug indications". *Nature Genetics* 47:8, 2015, pp. 856–860. DOI: 10.1038/ng.3314. PMID: 26121088. URL: https://doi.org/10.1038%2Fng.3314.

148. L. S. Nguyen, M. Vautier, Y. Allenbach, N. Zahr, O. Benveniste, C. Funck-Brentano, and J.-E. Salem. "Sirolimus and mTOR Inhibitors: A Review of Side Effects and Specific Management in Solid Organ Transplantation". *Drug Safety*, 2019. DOI: 10.1007/s40264-019-00810-9. PMID: 30868436. URL: https://doi.org/10.1007%2Fs40264-019-00810-9.

149. D. Nicholas, C. Boukacem-Zeghmouri, B. Rodríguez-Bravo, J. Xu, A. Watkinson, A. Abrizah, E. Herman, and M. Świgoń. "Where and how early career researchers find scholarly information". *Learned Publishing* 30:1, 2017, pp. 19–29. DOI: 10.1002/leap.1087. URL: https://doi.org/10.1002%5C%2Fleap.1087.

150. C. D. Nichols. "Drosophila melanogaster neurobiology, neuropharmacology, and how the fly can inform central nervous system drug discovery". *Pharmacology and Therapeutics* 112:3, 2006, pp. 677–700. DOI: 10.1016/j.pharmthera.2006.05.012. PMID: 16935347. URL: https://doi.org/10.1016%2Fj.pharmthera.2006.05.012.

151. X. Niu and B. M. Hemminger. "A study of factors that affect the information-seeking behavior of academic scientists". *Journal of the American Society for Information Science and Technology* 63:2, 2011, pp. 336–353. DOI: 10.1002/asi.21669. URL: https://doi.org/10.1002%5C%2Fasi.21669.

152. W. S. Noble. "How does multiple testing correction work?" *Nature biotechnology* 27:12, 2009, pp. 1135–1137.

153. H.-W. Nützmann, A. Huang, and A. Osbourn. "Plant metabolic clusters – from genetics to genomics". *New Phytologist* 211:3, 2016, pp. 771–789. DOI: 10.1111/nph.13981. pmcid: PMC5449196. URL: https://doi.org/10.1111%2Fnph.13981.

154. H.-W. Nützmann and A. Osbourn. "Gene clustering in plant specialized metabolism". *Current Opinion in Biotechnology* 26, 2014, pp. 91–99. DOI: 10.1016/j.copbio.2013.10.009. PMID: 24679264. URL: https://doi.org/10.1016%2Fj.copbio.2013.10.009.

155. K. Nykamp, M. Anderson, M. Powers, J. Garcia, B. Herrera, Y.-Y. Ho, Y. Kobayashi, N. Patil, J. Thusberg, M. Westbrook, and S. Topper. "Sherloc: a comprehensive refinement of the ACMG–AMP variant classification criteria". *Genetics in Medicine* 19:10, 2017, pp. 1105–1117. DOI: 10.1038/gim.2017.37. pmcid: PMC5632818. URL: https://doi.org/10.1038%2Fgim.2017.37.

156. T. I. Oprea. "Exploring the dark genome: implications for precision medicine". *Mammalian Genome* 30:7-8, 2019, pp. 192–200. DOI: 10.1007/s00335-019-09809-0. pmcid: PMC6836689. URL: https://doi.org/10.1007%5C%2Fs00335-019-09809-0.

157. T. I. Oprea, L. Jan, G. L. Johnson, B. L. Roth, A. Ma'ayan, S. Schürer, B. K. Shoichet, L. A. Sklar, and M. T. McManus. "Far away from the lamppost". *PLOS Biology* 16:12, 2018, e3000067. DOI: 10.1371/journal.pbio.3000067. pmcid: PMC6289406. URL: https://doi.org/10.1371%5C%2Fjournal.pbio.3000067.

158. A. Osbourn. "Secondary metabolic gene clusters: evolutionary toolkits for chemical innovation". *Trends in Genetics* 26:10, 2010, pp. 449–457. DOI: 10.1016/j.tig.2010.07.001. PMID: 20739089. URL: https://doi.org/10.1016%2Fj.tig.2010.07.001.

159. R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. "The use of gene clusters to infer functional coupling". *Proceedings of the National Academy of Sciences* 96:6, 1999, pp. 2896–2901. DOI: 10.1073/pnas.96.6.2896. pmcid: PMC15866. URL: https://doi.org/10.1073%2Fpnas.96.6.2896.

160. L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, 1999. URL: http://ilpubs.stanford.edu:8090/422/.

161. L. Peña-Castillo and T. R. Hughes. "Why Are There Still Over 1000 Uncharacterized Yeast Genes?" *Genetics* 176:1, 2007, pp. 7–14. DOI: 10.1534/genetics.107.074468. pmcid: PMC1893027. URL: https://doi.org/10.1534%5C%2Fgenetics.107.074468.

162. J. Peng, H. Li, Q. Jiang, Y. Wang, and J. Chen. "An integrative approach for measuring semantic similarities using gene ontology". *BMC Systems Biology* 8:Suppl 5, 2014, S8. DOI: 10.1186/1752-0509-8-s5-s8. URL: https://doi.org/10.1186%5C%2F1752-0509-8-s5-s8.

163. J. Peng, H. Li, Y. Liu, L. Juan, Q. Jiang, Y. Wang, and J. Chen. "InteGO2: a web tool for measuring and visualizing gene semantic similarities using Gene Ontology". *BMC Genomics* 17:S5, 2016. DOI: 10.1186/s12864-016-2828-6. URL: https://doi.org/10.1186%5C%2Fs12864-016-2828-6.

164. J. Peng, G. Lu, H. Xue, T. Wang, and X. Shang. "TSGOE: A web tool for tissue-specific gene ontology enrichment". In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018. DOI: 10.1109/bibm.2018.8621204. URL: https://doi.org/10.1109%5C%2Fbibm.2018.8621204.

165. J. Peng, S. Uygun, T. Kim, Y. Wang, S. Y. Rhee, and J. Chen. "Measuring semantic similarities by combining gene ontology annotations and gene co-function networks". *BMC Bioinformatics* 16:1, 2015. DOI: 10.1186/s12859-015-0474-7. URL: https://doi.org/10.1186%5C%2Fs12859-015-0474-7.

166. J. Peng, X. Wang, and X. Shang. "Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-Seq data". *BMC Bioinformatics* 20:S8, 2019. DOI: 10.1186/s12859-019-2769-6. URL: https://doi.org/10.1186%5C%2Fs12859-019-2769-6.

167. J. Peng, X. Zhang, W. Hui, J. Lu, Q. Li, S. Liu, and X. Shang. "Improving the measurement of semantic similarity by combining gene ontology and co-functional network: a random walk based approach". *BMC Systems Biology* 12:S2, 2018. DOI: 10.1186/s12918-018-0539-0. URL: https://doi.org/10.1186%5C%2Fs12918-018-0539-0.

168. L. A. Pennacchio, W. Bickmore, A. Dean, M. A. Nobrega, and G. Bejerano. "Enhancers: five essential questions". *Nature Reviews Genetics* 14:4, 2013, pp. 288–295. DOI: 10.1038/nrg3458. pmcid: PMC4445073. URL: https://doi.org/10.1038%2Fnrg3458.

169. C. Pesquita, D. Faria, H. Bastos, A. E. Ferreira, A. O. Falcão, and F. M. Couto. "Metrics for GO based protein semantic similarity: a systematic evaluation". *BMC Bioinformatics* 9:S5, 2008. DOI: 10.1186/1471-2105-9-s5-s4. URL: https://doi.org/10.1186%5C%2F1471-2105-9-s5-s4.

170. J. Piñero, J. M. Ramırez-Anguita, J. Saüch-Pitarch, F. Ronzano, E. Centeno, F. Sanz, and L. I. Furlong. "The DisGeNET knowledge platform for disease genomics: 2019 update". *Nucleic Acids Research*, 2019. DOI: 10.1093/nar/gkz1021. pmcid: PMC7145631. URL: https://doi.org/10.1093%2Fnar%2Fgkz1021.

171. A. d. Pozo, F. Pazos, and A. Valencia. "Defining functional distances over Gene Ontology". *BMC Bioinformatics* 9:1, 2008. DOI: 10.1186/1471-2105-9-50. URL: https://doi.org/10.1186%5C%2F1471-2105-9-50.

172. A. Purmann, J. Toedling, M. Schueler, P. Carninci, H. Lehrach, Y. Hayashizaki, W. Huber, and S. Sperling. "Genomic organization of transcriptomes in mammals: Coregulation and cofunctionality". *Genomics* 89:5, 2007, pp. 580–587. DOI: 10.1016/j.ygeno.2007.01.010. PMID: 17369017. URL: https://doi.org/10.1016%2Fj.ygeno.2007.01.010.

173. A. R. Quinlan and I. M. Hall. "BEDTools: a flexible suite of utilities for comparing genomic features". *Bioinformatics* 26:6, 2010, pp. 841–842. DOI: 10.1093/bioinformatics/btq033. pmcid: PMC2832824. URL: https://doi.org/10.1093%2Fbioinformatics%2Fbtq033.

174. K. L. Reddy, J. M. Zullo, E. Bertolino, and H. Singh. "Transcriptional repression mediated by repositioning of genes to the nuclear lamina". *Nature* 452:7184, 2008, pp. 243–247. DOI: 10.1038/nature06727. PMID: 18272965. URL: https://doi.org/10.1038%2Fnature06727.

175. L. T. Reiter. "A Systematic Analysis of Human Disease-Associated Gene Sequences In Drosophila melanogaster". *Genome Research* 11:6, 2001, pp. 1114–1125. DOI: 10.1101/gr.169101. pmcid: PMC311089. URL: https://doi.org/10.1101%2Fgr.169101.

176. P. Resnik. "Using information content to evaluate semantic similarity in a taxonomy". *arXiv preprint cmp-lg/9511007*, 1995.

177. P. Resnik. "Using information content to evaluate semantic similarity in a taxonomy". *arXiv preprint cmp-lg/9511007*, 1995.

178. L. E. Rieder and E. N. Larschan. "Wisdom from the fly". *Trends in Genetics* 30:11, 2014, pp. 479–481. DOI: 10.1016/j.tig.2014.08.003. pmcid: PMC4906897. URL: https://doi.org/10.1016%2Fj.tig.2014.08.003.

179. I. Rivals, L. Personnaz, L. Taing, and M.-C. Potier. "Enrichment or depletion of a GO category within a class of genes: which test?" *Bioinformatics* 23:4, 2006, pp. 401–407.

180. G. Rodgers, C. Austin, J. Anderson, A. Pawlyk, C. Colvis, R. Margolis, and J. Baker. "Glimmers in illuminating the druggable genome". *Nature Reviews Drug Discovery* 17:5, 2018, pp. 301–302. DOI: 10.1038/nrd.2017.252. pmcid: PMC6309548. URL: https://doi.org/10.1038%5C%2Fnrd.2017.252.

181. X. Ruan. "Long Non-Coding RNA Central of Glucose Homeostasis". *Journal of Cellular Biochemistry* 117:5, 2015, pp. 1061–1065. DOI: 10.1002/jcb.25427. PMID: 26530464. URL: https://doi.org/10.1002%2Fjcb.25427.

182. A. P. Russ and S. Lampel. "The druggable genome: an update". *Drug Discovery Today* 10:23-24, 2005, pp. 1607–1610. DOI: 10.1016/S1359-6446(05)03666-4. PMID: 16376820. URL: https://doi.org/10.1016%5C%2Fs1359-6446%5C%2805%5C%2903666-4.

183. P. H. Russell, R. L. Johnson, S. Ananthan, B. Harnke, and N. E. Carlson. "A large-scale analysis of bioinformatics code on GitHub". *PLOS ONE* 13:10, 2018. Ed. by Z. Qin, e0205898. DOI: 10.1371/journal.pone.0205898. pmcid: PMC6209220. URL: https://doi.org/10.1371%2Fjournal.pone.0205898.

184. D. Sánchez, M. Batet, D. Isern, and A. Valls. "Ontology-based semantic similarity: A new feature-based approach". *Expert Systems with Applications* 39:9, 2012, pp. 7718–7728. DOI: 10.1016/j.eswa.2012.01.082. URL: https://doi.org/10.1016%5C%2Fj.eswa.2012.01.082.

185. R. Santos, O. Ursu, A. Gaulton, A. P. Bento, R. S. Donadi, C. G. Bologa, A. Karlsson, B. Al-Lazikani, A. Hersey, T. I. Oprea, and J. P. Overington. "A comprehensive map of molecular drug targets". *Nature Reviews Drug Discovery* 16:1, 2016, pp. 19–34. DOI: 10.1038/nrd.2016.230. URL: https://doi.org/10.1038%5C%2Fnrd.2016.230.

186. E. W. Sayers, R. Agarwala, E. E. Bolton, J. Brister, K. Canese, K. Clark, R. Connor, N. Fiorini, K. Funk, T. Hefferon, J. Holmes, S. Kim, A. Kimchi, P. A. Kitts, S. Lathrop, Z. Lu, T. L. Madden, A. Marchler-Bauer, L. Phan, V. A. Schneider, C. L. Schoch, K. D. Pruitt, and J. Ostell. "Database resources of the National Center for Biotechnology Information". *Nucleic Acids Research* 47:D1, 2018, pp. D23–D28. DOI: 10.1093/nar/gky1069. URL: https://doi.org/10.1093%5C%2Fnar%5C%2Fgky1069.

187. A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer. "A new measure for functional similarity of gene products based on Gene Ontology". *BMC Bioinformatics* 7:1, 2006. DOI: 10.1186/1471-2105-7-302. URL: https://doi.org/10.1186%5C%2F1471-2105-7-302.

188. T. Schwecke, J. F. Aparicio, I. Molnar, A. Konig, L. E. Khaw, S. F. Haydock, M. Oliynyk, P. Caffrey, J. Cortes, and J. B. Lester. "The biosynthetic gene cluster for the polyketide immunosuppressant rapamycin." *Proceedings of the National Academy of Sciences* 92:17, 1995, pp. 7839–7843. DOI: 10.1073/pnas.92.17.7839. pmcid: PMC41241. URL: https://doi.org/10.1073%2Fpnas.92.17.7839.

189. S. Seabold and J. Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

190. T. Shiina, H. Inoko, and J. K. Kulski. "An update of the HLA genomic region, locus information and disease associations: 2004". *Tissue Antigens* 64:6, 2004, pp. 631–649. DOI: 10.1111/j.1399-0039.2004.00327.x. PMID: 15546336. URL: https://doi.org/10.1111%2Fj.1399-0039.2004.00327.x.

191. D. Shlyueva, G. Stampfel, and A. Stark. "Transcriptional enhancers: from properties to genome-wide predictions". *Nature Reviews Genetics* 15:4, 2014, pp. 272–286. DOI: 10.1038/nrg3682. PMID: 24614317. URL: https://doi.org/10.1038%2Fnrg3682.

192. M. Sipiczki. *Genome Biology* 1:2, 2000, reviews1011.1. DOI: 10.1186/gb-2000-1-2-reviews1011. pmcid: PMC138848. URL: https://doi.org/10.1186%5C%2Fgb-2000-1-2-reviews1011.

193. B. Snel. "STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene". *Nucleic Acids Research* 28:18, 2000, pp. 3442–3444. DOI: 10.1093/nar/28.18.3442. pmcid: PMC110752. URL: https://doi.org/10.1093%2Fnar%2F28.18.3442.

194. M. A. G. Sosa, R. D. Gasperi, and G. A. Elder. "Modeling human neurodegenerative diseases in transgenic systems". *Human Genetics* 131:4, 2011, pp. 535–563. DOI: 10.1007/s00439-011-1119-1. PMID: 22167414. URL: https://doi.org/10.1007%2Fs00439-011-1119-1.

195. P. T. Spellman and G. M. Rubin. *Journal of Biology* 1:1, 2002, p. 5. DOI: 10.1186/1475-4924-1-5. pmcid: PMC117248. URL: https://doi.org/10.1186%2F1475-4924-1-5.

196. J. Spieth, G. Brooke, S. Kuersten, K. Lea, and T. Blumenthal. "Operons in C. elegans: Polycistronic mRNA precursors are processed by trans-splicing of SL2 to downstream coding regions". *Cell* 73:3, 1993, pp. 521–532. DOI: 10.1016/0092-8674(93)90139-h. URL: https://doi.org/10.1016%2F0092-8674%2893%2990139-h.

197. D. Sproul, N. Gilbert, and W. A. Bickmore. "The role of chromatin structure in regulating the expression of clustered genes". *Nature Reviews Genetics* 6:10, 2005, pp. 775–781. DOI: 10.1038/nrg1688. PMID: 16160692. URL: https://doi.org/10.1038%2Fnrg1688.

198. T. Stoeger, M. Gerlach, R. I. Morimoto, and L. A. N. Amaral. "Large-scale investigation of the reasons why potentially important genes are ignored". *PLOS Biology* 16:9, 2018. Ed. by T. Freeman, e2006643. DOI: 10.1371/journal.pbio.2006643. pmcid: PMC6143198. URL: https://doi.org/10.1371%2Fjournal.pbio.2006643.

199. A. Sturtevant. "The Effects of Unequal Crossing over at the Bar Locus in Drosophila". *Genetics* 2:10, 1925, pp. 117–147. pmcid: PMC1200852. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1200852/pdf/117.pdf.

200. P. H. Sudmant, M. S. Alexis, and C. B. Burge. "Meta-analysis of RNA-seq expression data across species, tissues and studies". *Genome Biology* 16:1, 2015. DOI: 10.1186/s13059-015-0853-4. pmcid: PMC4699362. URL: https://doi.org/10.1186%2Fs13059-015-0853-4.

201. F. Supek, M. Bošnjak, N. Škunca, and T. Šmuc. "REVIGO summarizes and visualizes long lists of gene ontology terms". *PloS one* 6:7, 2011, e21800.

202. F. Supek and N. Škunca. "Visualizing GO Annotations". *The Gene Ontology Handbook*, 2017, pp. 207–220.

203. S. Swaminathan, D. Morrone, Q. Wang, D. B. Fulton, and R. J. Peters. "CYP76M7 Is an ent-Cassadiene C11-alpha-Hydroxylase Defining a Second Multifunctional Diterpenoid Biosynthetic Gene Cluster in Rice". *The Plant Cell* 21:10, 2009, pp. 3315–3325. DOI: 10.1105/tpc.108.063677. pmcid: PMC2782285. URL: https://doi.org/10.1105%2Ftpc.108.063677.

204. E. Szél, R. Bozó, É. Hunyadi-Gulyás, M. Manczinger, K. Szabó, L. Kemény, Z. Bata-Csörgő, and G. Groma. "Comprehensive Proteomic Analysis Reveals Intermediate Stage of Non-Lesional Psoriatic Skin and Points out the Importance of Proteins Outside this Trend". *Scientific Reports* 9:1, 2019. DOI: 10.1038/s41598-019-47774-5. pmcid:

PMC6684579. URL:
https://doi.org/10.1038%2Fs41598-019-47774-5.

205. K.-Y. Teng and K. Ghoshal. "Role of Noncoding RNAs as Biomarker and Therapeutic Targets for Liver Fibrosis". *Gene Expression* 16:4, 2015, pp. 155–162. DOI: 10.3727/105221615X14399878166078. pmcid: PMC4689200. URL:
https://doi.org/10.3727%2F105221615x14399878166078.

206. "The Gene Ontology Resource: 20 years and still GOing strong". *Nucleic Acids Research* 47:D1, 2018, pp. D330–D338.
DOI: 10.1093/nar/gky1055. URL:
https://doi.org/10.1093%5C%2Fnar%5C%2Fgky1055.

207. D. Turnbull. *Relevant search : with applications for Solr and Elasticsearch*. Manning Publications Co, Shelter Island, NY, 2016. ISBN: 9781617292774.

208. S. Verschuere, N. Navassiolava, L. Martin, P. I. Nevalainen, P. J. Coucke, and O. M. Vanakker. "Reassessment of causality of ABCC6 missense variants associated with pseudoxanthoma elasticum based on Sherloc". *Genetics in Medicine* 23:1, 2020, pp. 131–139.
DOI: 10.1038/s41436-020-00945-6. PMID: 32873932. URL:
https://doi.org/10.1038%2Fs41436-020-00945-6.

209. C. VÉZINA, A. KUDELSKI, and S. N. SEHGAL. "Rapamycin (AY-22,989), a new antifungal antibiotic. I. Taxonomy of the producing streptomycete and isolation of the active principle." *The Journal of Antibiotics* 28:10, 1975, pp. 721–726.
DOI: 10.7164/antibiotics.28.721. PMID: 1102508. URL:
https://doi.org/10.7164%2Fantibiotics.28.721.

210. L. Wadi, M. Meyer, J. Weiser, L. D. Stein, and J. Reimand. "Impact of outdated gene annotations on pathway enrichment analysis". *Nature Methods* 13, 2016, p. 705. DOI: 10.1038/nmeth.3963. URL:
http://dx.doi.org/10.1038/nmeth.3963.

211. M. F. Wangler, S. Yamamoto, and H. J. Bellen. "Fruit Flies in Biomedical Research". *Genetics* 199:3, 2015, pp. 639–653.
DOI: 10.1534/genetics.114.171785. pmcid: PMC4349060. URL:
https://doi.org/10.1534%2Fgenetics.114.171785.

212. R. W. White and R. A. Roth. "Exploratory Search: Beyond the Query-Response Paradigm". *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1:1, 2009, pp. 1–98.
DOI: 10.2200/s00174ed1v01y200901icr003. URL: https://doi.org/10.2200%5C%2Fs00174ed1v01y200901icr003.

213. B. Wichor Matthijs Bramer. "Variation in number of hits for complex searches in Google Scholar". *Journal of the Medical Library Association* 104:2, 2016. DOI: 10.5195/jmla.2016.61. URL: https://doi.org/10.5195%5C%2Fjmla.2016.61.

214. M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. d. S. Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. '. Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons. "The FAIR Guiding Principles for scientific data management and stewardship". *Scientific Data* 3:1, 2016. DOI: 10.1038/sdata.2016.18. pmcid: PMC4792175. URL: https://doi.org/10.1038%5C%2Fsdata.2016.18.

215. M. F. Wolfner and D. E. Miller. "Alfred Sturtevant Walks into a Bar: Gene Dosage, Gene Position, and Unequal Crossing Over in Drosophila". *Genetics* 204:3, 2016, pp. 833–835. DOI: 10.1534/genetics.116.195891. pmcid: PMC5105861. URL: https://doi.org/10.1534%2Fgenetics.116.195891.

216. V. Wood, A. Lock, M. A. Harris, K. Rutherford, J. Bähler, and S. G. Oliver. "Hidden in plain sight: what remains to be discovered in the eukaryotic proteome?" *Open Biology* 9:2, 2019, p. 180241. DOI: 10.1098/rsob.180241. pmcid: PMC6395881. URL: https://doi.org/10.1098%5C%2Frsob.180241.

217. H. Yang, T. Nepusz, and A. Paccanaro. "Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty". *Bioinformatics* 28:10, 2012, pp. 1383–1389. DOI: 10.1093/bioinformatics/bts129. URL: https://doi.org/10.1093%5C%2Fbioinformatics%5C%2Fbts129.

218. F. Yue et al. "A comparative encyclopedia of DNA elements in the mouse genome". *Nature* 515:7527, 2014, pp. 355–364. DOI: 10.1038/nature13992. pmcid: PMC4266106. URL: https://doi.org/10.1038%2Fnature13992.

219. X.-Y. Zhao and J. D. Lin. "Long Noncoding RNAs: A New Regulatory Code in Metabolic Control". *Trends in Biochemical Sciences* 40:10, 2015, pp. 586–596. DOI: 10.1016/j.tibs.2015.08.002. pmcid: PMC4584418. URL: https://doi.org/10.1016%2Fj.tibs.2015.08.002.

Appendix A: Peer-reviewed papers and thesis chapters

Figure A.1 shows how my three peer-reviewed papers informed the research questions that arose while working on this thesis. The papers include a gene ontology (GO) paper [106], a literature search paper [105], and a mouse paper [119].

A.1 Gene ontology paper

The work done on my gene ontology paper [106] addressed open Gene Ontology Enrichment Analysis (GOEA) questions:

- How correct are the GOEA results?
- How many expected Gene Ontology (GO) results are missing?
- What affects how many results are missing?
- Should I use the "propagate counts" mode?
- How do I summarize the GO results?
- How do I visualize the GO results?

After conducting the research required to write this paper, I felt confident about the portions of my thesis that use GOEAs.

A.2 Literature search paper

Knowledge acquired through work on the literature search paper [105] helped inform the scant literature search results at the beginning of my thesis. It was interesting to find that many highly esteemed authors that appeared in the final literature search of the thesis did not appear in the early search using Google Scholar.

Scientific literature search results should be reproducible [22] [80] [79]. PubMed provides reproducible literature searches, while Google Scholar does not. My literature search results for this thesis have dramatically improved through my PhD time by combining citation data from the National Institutes of Health Open Citation Collection [92], the PubMed web search experience, programmatic access to PubMed's data using NCBI's E-Utils libraries [146], and revision management of the literature search results using git, all coordinated with my Python package [105] found at https://github.com/dvklopfenstein/pmidcite.

A.3 Mouse paper

Any analysis or plot in this thesis can also be performed with mice and flies and can be expanded to additional species. My work on the Cyclin-D1/G9a paper [119] provided practical data for working with the mouse genome and comparing mouse clusters to clusters in the human genome. It also provided experience with comparing LAD region location data annotated on an older mouse genome to the cyclin-D1/G9a molecule binding point on a newer mouse genome.

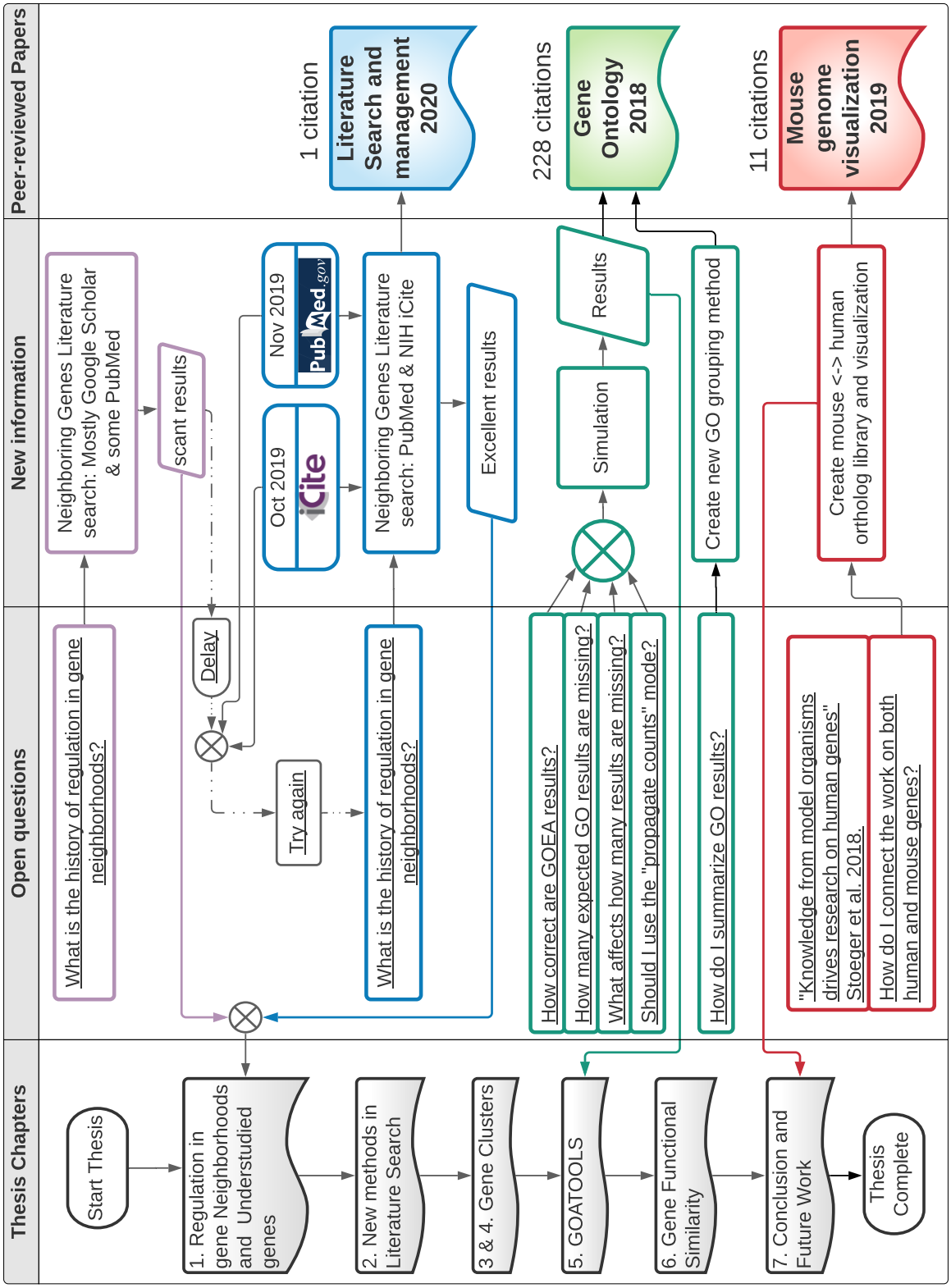

Figure A.1: **How the peer-reviewed papers fit into the thesis.** The PhD thesis chapters appear as gray shaded boxes in the left-most column. The peer-reviewed papers appear as blue, green, and red shaded boxes in the right-most column. Citation counts are from Google Scholar in June 2021. The the research question process flow is in the middle columns.

Appendix B: The GOA TOOLs paper

B.1 Over 1,700 bioinformatics repos ranked

In 2018, Russell et al. defined the impact of individual open-source bioinformatics projects on the research community [183]. They searched for peer-reviewed research papers describing bioinformatics projects hosted on GitHub, finding 1,720 such projects. They found that the bottom 28% (482) of the papers were not cited by any articles in PubMed Central, 40% (688) were cited by 1 to 5 articles, and that 32% (550) of the papers have more than five citations.

Russell et al. highlighted 23 projects out of 1,720 as being particularly high profile and crucial in the research community. One of the open-source projects featured in this thesis, GOA TOOLs, sits among the top 1% of high-profile projects, such as samtools [40] and bedtools [173], based on the parameters identified by Russell of being associated with a impactful project, as shown in Figures B.1 and B.2.

GOA TOOLs is a Python library and tool suite for managing gene ontology (GO) terms and running gene ontology enrichment analyses (GOEA) and is freely available at https://github.com/tanghaibao/goatools. GO terms describe the biological functions of a gene product.

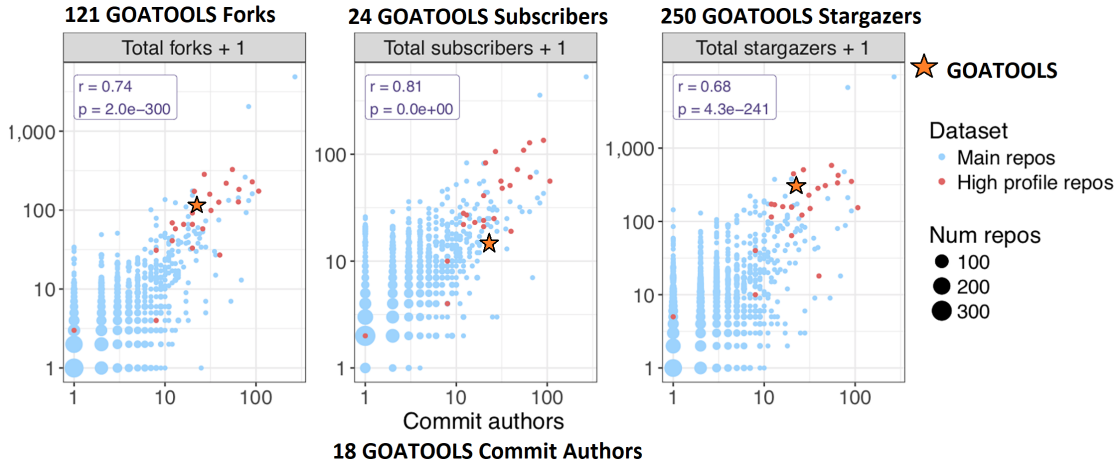

Figure B.1: **GOA TOOLs is a top project based on three parameters from data collected by GitHub.** Russell found that the three GitHub features shown here correlate to an impactful bioinformatics project. The x and y axes for each of the three panels are in a log 10 scale, resulting in over several hundred projects in the lower left-hand corner and only tens of high-profile projects in the upper right-hand corner. GOA TOOLs is comfortably placed in the upper right-hand corner. A "fork" occurs in GitHub when another researcher makes a copy of a project. A "watcher" is a researcher who signs up for email when there are project discussions. A "stargazer" is a researcher who has signaled that they like the project by "starring" it in GitHub. This figure was adapted with permission from Figure 4 in Russell et al. [183] using GOA TOOLs GitHub data from 2018, the same publication year of Russell's paper.

Repo Name	Lang.	Mb	Stars	Forks	Watchers
biopython	Python	49	1761	914	144
samtools	C	12	762	385	107
bedtools2	C	47	526	210	64
igv	Java	410	342	162	48
GOATOOLS	Python	34	250	121	24
bowtie2	C++	153	240	82	28
vcftools	C++	1	225	96	27
bcftools	C	14	211	122	48
cufflinks	C++	491	193	101	42
htsjdk	Java	72	184	209	50
tophat	C++	11	80	38	19

Figure B.2: **GOA TOOLS is in the top 1% of bioinformatics projects.** GOA TOOLs sits comfortably among the top 23 of 1,720 bioinformatics projects, such as samtools [40], bedtools [173], and tophat [103], when using the criteria determined by Russel et al. to correlate to an impac tful bioinformatics project. Important criteria include the size of the project, the number of GitHub stars given by the research community, and other GitHub features such as forks and watchers. The data for this table are from the supplemental data of Russell et al. [183]. This table uses GOA TOOLs GitHub data from 2018, the same publication year of Russell's paper.

B.2 Researcher contributions and community interest

Researchers show their appreciation for an open-source project on GitHub by awarding it stars. The rate of researcher appreciation of the GOA TOOLs project increased substantially between the beginning of the project in January 2010 (when it was awarded its first star) and the publication date of this thesis in June 2021 (with 432 stars).

Table B.1 shows that GOA TOOLs has three distinct appreciation periods divided by two events. The first event in 2016 marks the beginning of my adding functionality to GOA TOOLs for the paper while writing the paper (Table B.1: b). The second event is the publication of the paper in the journal *Scientific Reports* in 2018 [106] (Table B.1: c).

Table B.1: **The rate of researcher appreciation rose 1,000% following D.V. Klopfenstein's contributions.** The letters a, b, c, and d in the start time and end time columns indicate consequential dates in the GOA TOOLs project. There are three rating periods in the table. The GOA TOOLs project was awarded 2.47 stars per 100 days before my major code contributions. The rate increased by 500% from 2.47 to 12.42 when I began committing new functionality for the GOA TOOLs paper. The rate increased 1,000% from 2.47 to 25.07 after the publication of the paper.

Stars per 100 days	Start time	End time	Time period description
2.47	Jan 2010 (a)	Jan 2016 (b)	Beginning of project to paper proposal
12.42	Jan 2016 (b)	Jul 2018 (c)	Paper proposal to paper publication
25.07	Jul 2018 (c)	Jun 2021 (d)	Paper publication to June 2021

The rate of appreciation increased fivefold from 2.47 to 12.42 stars/100 days after I proposed to the project's owner, Dr. Haibao Tang, that I write a

peer-reviewed research paper centering on GOA TOOLs and began adding the functionality needed for the paper. The rate of appreciation doubled from 12.42 to 25.07 stars/100 days after the publication of the paper for a total tenfold increase the original appreciation rating of 2.47.

Figure B.3 shows researcher interest in GOA TOOLs (top panel) along with my contributions (bottom panel). In the top panel, researcher interest is shown using short vertical blue lines to represent the date when researchers added stars to the GOA TOOLs project. The three rating periods are divided by the dashed magenta lines (Figure B.3: b and c).

The bottom panel shows my additions of code, called commits, as the principal contributor out of a total of 19 GOA TOOLs contributors. Plots of the code contributions from all 19 contributors are available on the GOA TOOLs website (https://github.com/tanghaibao/goatools/graphs/contributors).

The GitHub stars rose from 55 (Figure B.3: b) to 169 (Figure B.3: c) solely due to researchers noticing my new functionality in GitHub. There was no announcement of any improvements using any channels such as papers or social media posts. Researchers noticed the new functionality and immediately responded by quintupling the rate of stars awarded.

Dr. Tang agreed to my adding the functionality for the paper in January 2016. My first major commits in support of the paper (Figure B.3: U) show my first contributions of the new functionality to GOA TOOLs that was created for this thesis.

For example, my advisor wanted the multiple GOEA *p*-values to be corrected using the Benjamini-Hochberg multiple test correction, which was not available in GOA TOOLs before 2016. Consequently, I added support for

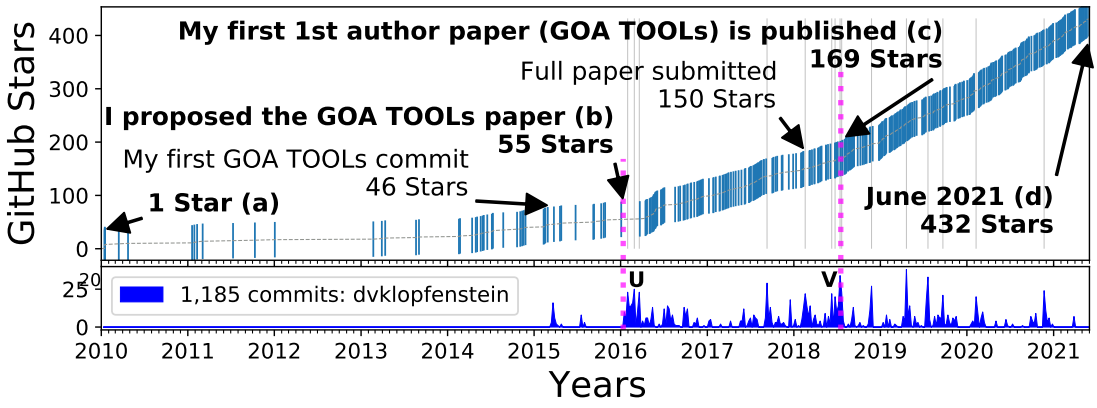

Figure B.3: **The amount of researcher interest in GOA TOOLs rose rapidly following the creation of commits by D. V. Klopfenstein starting in 2016.** The top panel shows the interest of researchers in GOA TOOLs. The bottom panel shows the code contributions from the principal lead, D. V. Klopfenstein. The text and arrows in the top panel describe key project dates. The dotted magenta lines divide the figure into three time periods. The bold text ending with the letters a, b, c, and d mark consequential events (Table B.1). Each short vertical blue line in the top panel represents a date when a researcher added a star to GOA TOOLs, illustrating their appreciation of the project. The light gray lines in the top panel show the top 15 weeks of commit activity out of over 1,100 total commits by D. V. Klopfenstein. The x-axis is time in months and years and is shared by both panels. The y-axis of the top panel is the total number of GitHub stars given to the project, which in spring 2021 was over 400. The bottom panel shows new features contributed to GOA TOOLs by its top contributor. The number of commits appears on the y-axis. The letters U and V indicate periods of commit activity just after I suggested writing the paper (U) and just before the paper was published (V).

Benjamini-Hochberg, which is available through the statsmodels Python package [189], as well as access to all multiple test corrections available through the statsmodels package.

My collaborators wanted the GOEA results in an Excel spreadsheet format. I added the functionality (Figure B.3: U) while improving it by setting the default cell (column) widths to researcher-friendly values. For example, the column width is wider for lengthy columns like "name" and shorter for columns like "*p*-value."

The numerous contributions that I submitted in the first quarter of 2016 were followed by a jump in researcher interest as seen in the major acceleration of stars awarded to GOA TOOLs in the second quarter of 2016.

Because researchers expressed desire for additional functionality in GO plotting and because I used my augmented GO plotting to create figures for the paper, I released the improved GO plotting before publication, as shown by the spikes of commit activity in 2017 and early 2018 before the paper was published. I delayed submitting the novel functionality, which included an innovative novel GO grouping, until just before publication (Figure B.3: V).

After publication, I continued to add new functionalities to GOA TOOLs that were desired by the research community as expressed in the issues section of the project. Examples of new functionalities include robust support for the evidence codes which describe the proof (evidence) that a biological function is correctly attributed to the gene product to which it is annotated. Examples of evidence codes include the functionality determined by "experimental evidence" or "computationally determined." I also added support for using optional relationships between GO terms such as "part of" and "regulates" to augment the required relationship of "is a."

B.3 GOEA stochastic simulations

Dr. Tang suggested that I create original stochastic GOEA simulations and interpret the results in the paper to make the paper more captivating.

The stochastic GOEA simulation code and commits, results, data, and figures represent a project separate from the GOA TOOLs project and are freely available in the GitHub repository, (https://github.com/dvklopfenstein/goatools_simulation).

I began the simulation project by creating a more straightforward set of simulations, which include the Benjamini-Hochberg and Bonferroni multiple test corrections, to establish the proof-of-concept of the stochastic simulations; architect baseline implementations of simulation code; and explore methods to visualize the resulting statistical data in figures.

I built upon my successful deployment of the multiple test correction stochastic simulations to create the more complicated GOEA simulations. The simulation project uses the GOA TOOLs project as a prerequisite for the GOEA simulations. I am the sole contributor to the simulation repository.

The Curriculum Vita of D. V. Klopfenstein

dvklopfenstein@protonmail.com Philadelphia, PA, USA
ORCID: 0000-0003-0161-7603 *https://github.com/dvklopfenstein*

PROFESSIONAL PREPARATION

- Drexel University, Philadelphia, PA: PhD Biomedical Engineering
- Drexel University, Philadelphia, PA: MS Biomedical Engineering
- Rensselaer Polytechnic Institute, Troy, NY: BS Electrical Engineering

PEER-REVIEWED PUBLICATIONS

1. Commentary to Gusenbauer and Haddaway 2020: Evaluating retrieval qualities of Google Scholar and PubMed

2020, *Research Synthesis Methods*

- Wrote a commentary about another paper, a format that is by invitation-only of the editors.
- The authors from the original paper were invited to and did respond.
- Created a new method for managing a PubMed literature search.

2. GOATOOLS: A Python library for Gene Ontology (GO) analyses

2018, *Scientific Reports*

- Answered open questions by creating 100,000 stochastic simulations and then examining and interpreting the data.
- Dramatically expanded functionality in the open-source project:
  - Created a new method for grouping GO terms.
  - Since publishing, the GOA TOOLS open-source project is now trusted to be a prerequisite by ~90 other open-source projects.
  - Expanded research interest in GOATOOLS: GitHub stars rose from ~40 to over 430.
- Found a "showstopper" bug for the Gene Ontology Consortium's (GOC) annotation software, which they fixed.
- Found and reported numerous bugs in GO annotation to the GOC.

3. Cyclin D1 integrates G9a-mediated histone methylation

2019, *Oncogene*

Modified Jefferson's hypothesis:

- Created novel visualizations to examine data related to their hypothesis.
- Modified their hypothesis from "G9a/cyclin-D1 molecules binds to LAD regions" to "G9a/cyclin-D1 molecules binds to the edges of the LAD regions."
- Performed two statistical analyses to test the modified hypothesis.

FIRST PLACE POSTER AWARD

2014, *Sidney Kimmel Cancer Consortium Symposium* Won 1st Place Poster