BioInformatics coursework 2 - The Gene Curation Consortium (GenCC) Database of Gene-Disease Mappings

1. Introduction

In this report, we are going to explore biological databases, ontologies, and networks. We will particularly focus on the Gene Curation Consortium (GenCC) Database, which is a pivotal resource for understanding gene-disease mappings [1].

We will first perform a detailed analysis of a standard GenCC dataset, by exploring details including provenance information, submission dates and timelines, confidence categories, and the number of genes associated with each disease.

Afterwards, we are going to explore the concept of ***modes of inheritance*** (MOI), which describe the manner in which a genetic disorder is passed from one generation to the next [2]. For this part, we will be using another important bioinformatics resource, namely the Human Phenotype Ontology (HPO). HPO is a standardized vocabulary containing phenotypic information about genes or product genes [3].

Lastly, we will consider subsets of the original GenCC dataset previously used and explore gene-disease connections by combining these subsets with yet another powerful bioinformatics tool, namely the MONDO Disease Ontology. MONDO provides a logic-based structure for unifying multiple disease resources, with the aim of integrate the classifications and relationships of commonly used disease ontologies into a single resource [4]. We will then use these subsets to build a simple disease network based on genes associated in common between pairs of diseases, cluster this network and show the resulting graph.

2. Data & Methods

2.1 Data sources used

As part of this analysis, three main data sources were used. Below we provide some details about these three files:

1. GenCC flat-file: this file contains data from the Gene Curation Consortium (GenCC) Database, and the version used is from 30/10/2023.

2. HPO file: this file contains data from the Human Phenotype Ontology (HPO) and was downloaded from the NCBO BioPortal [5]. The version used is from 09/10/2023.

3. MONDO file: this file contains data from the MONDO Disease Ontology, and the version used is from 12/09/2023.

2.2 Methods used

For all four parts outlined below, python's Jupyter notebooks interface was the primary method used in this research for data analysis. However, below we are going to explore the individual

techniques used for each of the individual parts. We note that, for reproducibility purposes, we included all the code used for this study in the ***Appendix*** section.

Part 1

In this part, as previously mentioned, we performed a detailed analysis of the data from the GenCC flat-file. To do so, we used the ***pandas*** Python library to process the file and store it in a dataframe format. We then used the ***matplotlib*** and ***seaborn*** packages to create some visualizations to better analyze the data.

Part 2

Here we explored ***modes of inheritance*** (MOI) using the HPO file. We again used the ***pandas*** Python library, but we also used the ***pronto*** module in order to process the contents of the HPO file and conduct our analysis. Once again, the ***matplotlib*** and ***seaborn*** packages were used to create some visualizations to better analyze the data.

Part 3

For this part, we conducted three studies on the MONDO file. First, we looked at the term ***nervous system disorder*** and tried to identify diseases related to this given term as well as genes linked to those diseases. As part of the extension, we then performed the same analysis for two different terms, namely ***digestive system disorder*** and ***disorder of the visual system***. As in part 2, we used the ***pandas*** and ***pronto*** modules in order to process and analyze the data.

Part 4

Lastly, in this part we use the data from Part 3 to build a simple disease network based on genes associated in common between pairs of diseases, cluster this network and show the resulting graph. In order to do so, we again rely on the ***pandas*** Python library to process the data and use the ***matplotlib*** package for creating visualizations. However, we also use another module, namely ***networkx***, to build the actual disease network to be displayed. We note that, if the reader were to try to reproduce our results using the code in the Appendix, they might get slightly different results, as the outputted network changed with each iteration.

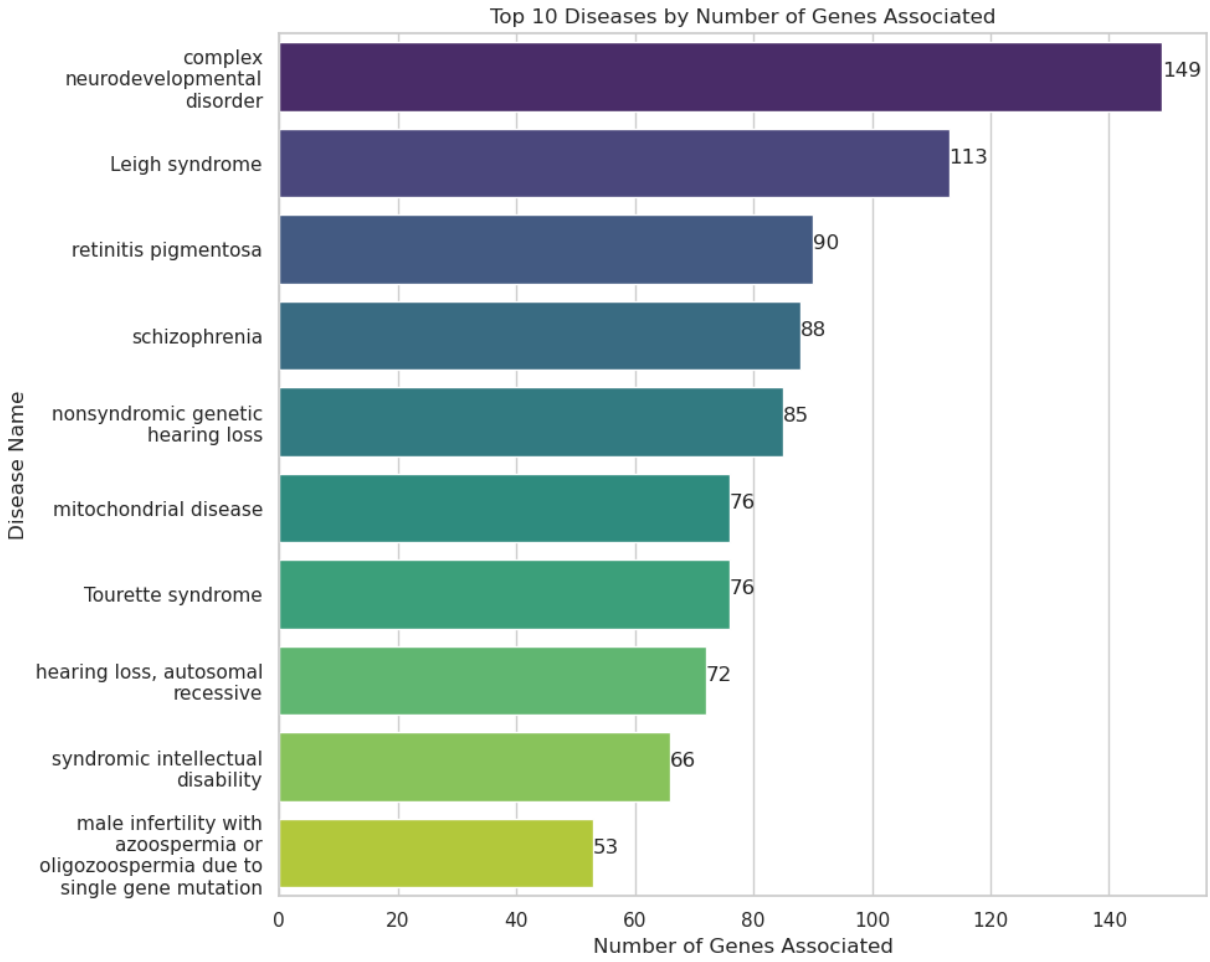
3. Results and analysis

Part One - Summary Analysis of GenCC Data

Task 1

Disease name	Corresponding number of genes
complex neurodevelopmental disorder	149
Leigh syndrome	113
retinitis pigmentosa	90
schizophrenia	88
nonsyndromic genetic hearing loss	85
mitochondrial disease	76
Tourette syndrome	76
hearing loss, autosomal recessive	72
syndromic intellectual disability	66
male infertility with azoospermia or oligozoospermia due to single gene mutation	53

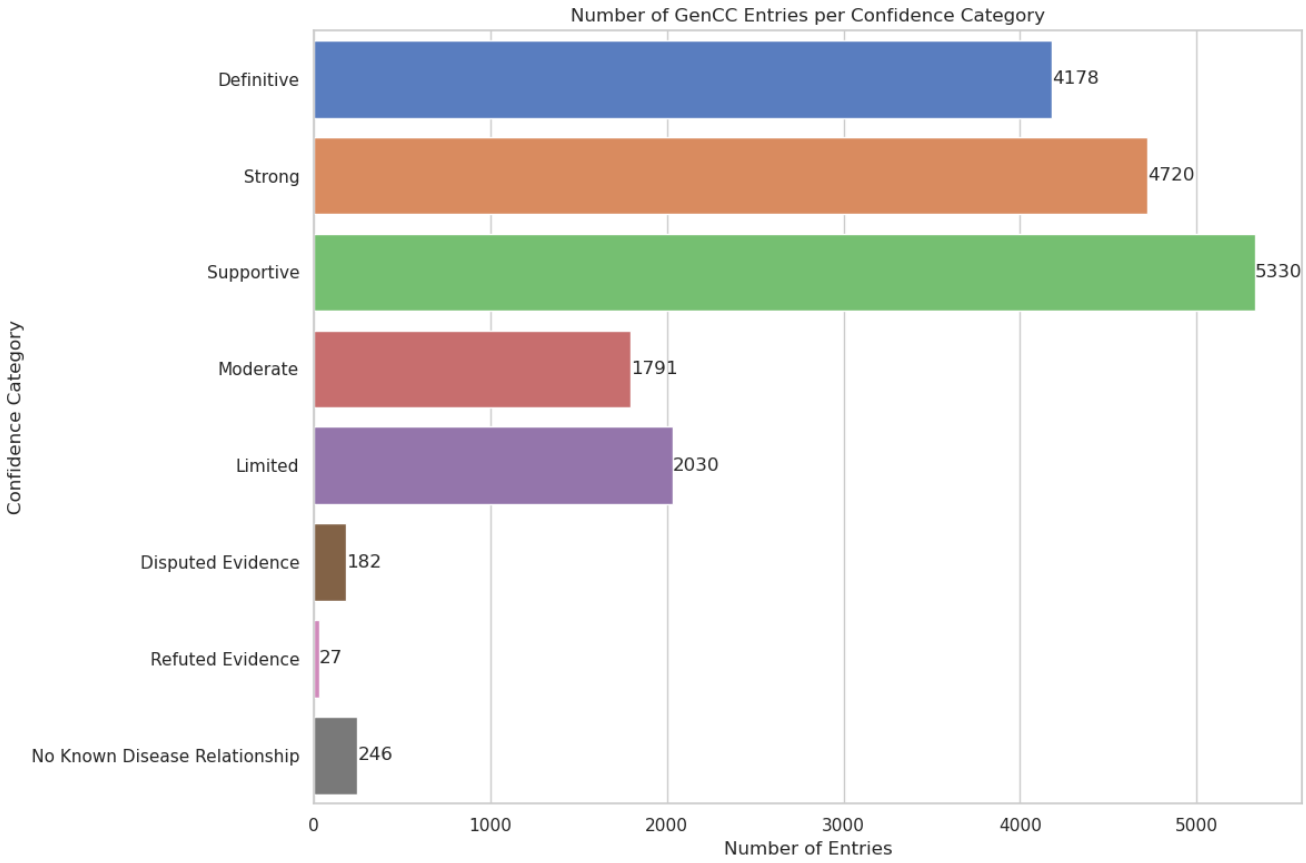
Task 2



Task 3

Confidence Category	Number of Entries
Definitive	4178
Strong	4720
Supportive	5330
Moderate	1791
Limited	2030
Disputed Evidence	182
Refuted Evidence	27
No Known Disease Relationship	246

Task 4



Task 5

Provenance Category	Number of Entries
PubMed ID	1706
DOI	533
Provenance Data (not PubMed or DOI)	15017
No Provenance Data	1248

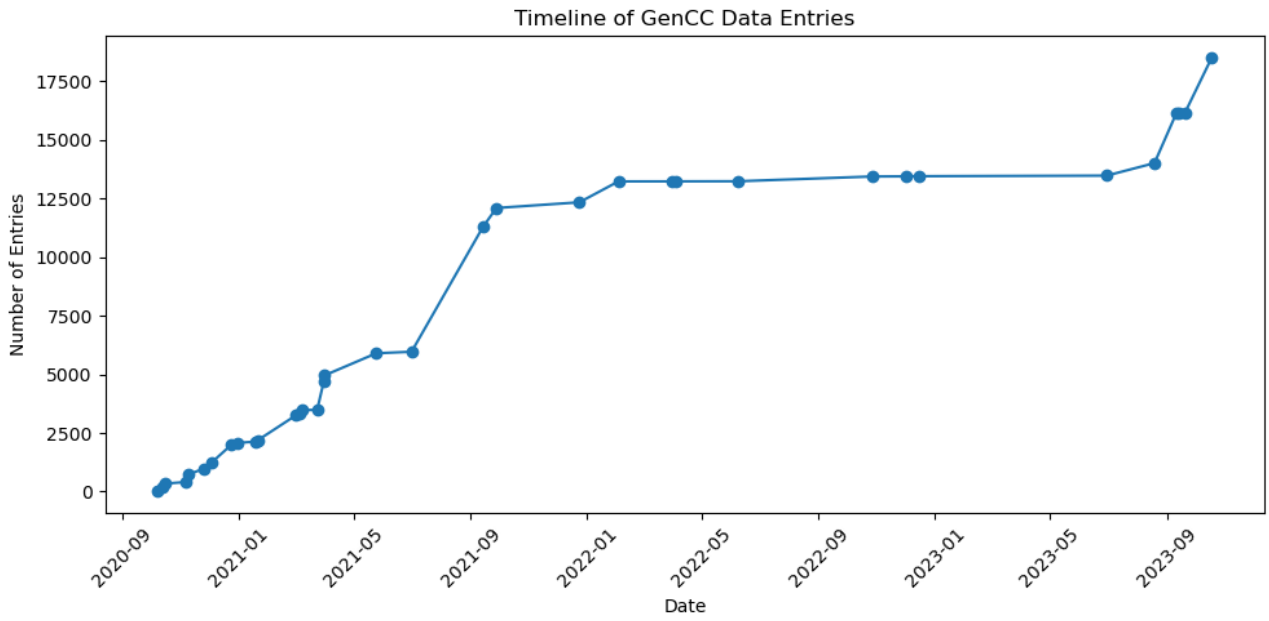
Analysis of the results

One thing worth noting is that, for the top 10 diseases by number of gene entries, all diseases have at least 50 corresponding genes; however, the range of the corresponding number of genes is fairly wide, with the first entry having almost triple the number of entries of the tenth entry.

Another aspect worth noting is that a good proportion of the data (around 77%) are reported with a confidence category of 'Supportive' or higher. This suggests that the data used for our study mostly comes from trusted sources, and any findings can be reported with fairly high confidence.

Lastly, we note that, although not much of the data has a PubMed ID or a DOI in terms of provenance, the vast majority does have some form of provenance, with only 1248 entries (around 9%) being labeled as having 'No Provenance Data'.

Extension



As we can see from the figure above, there is a general upwards trend in the number of GenCC data entries over time, which goes to show the widespread adoption of the GenCC platform by the biology community. We note in particular a sharp rise around January 2021, with this rise continuing almost until around November 2021. This increase could be due to the COVID-19 pandemic, as extensive research had been done in that specific area in order to provide solutions for combatting the spread of the virus.

Part Two - Modes of Inheritance

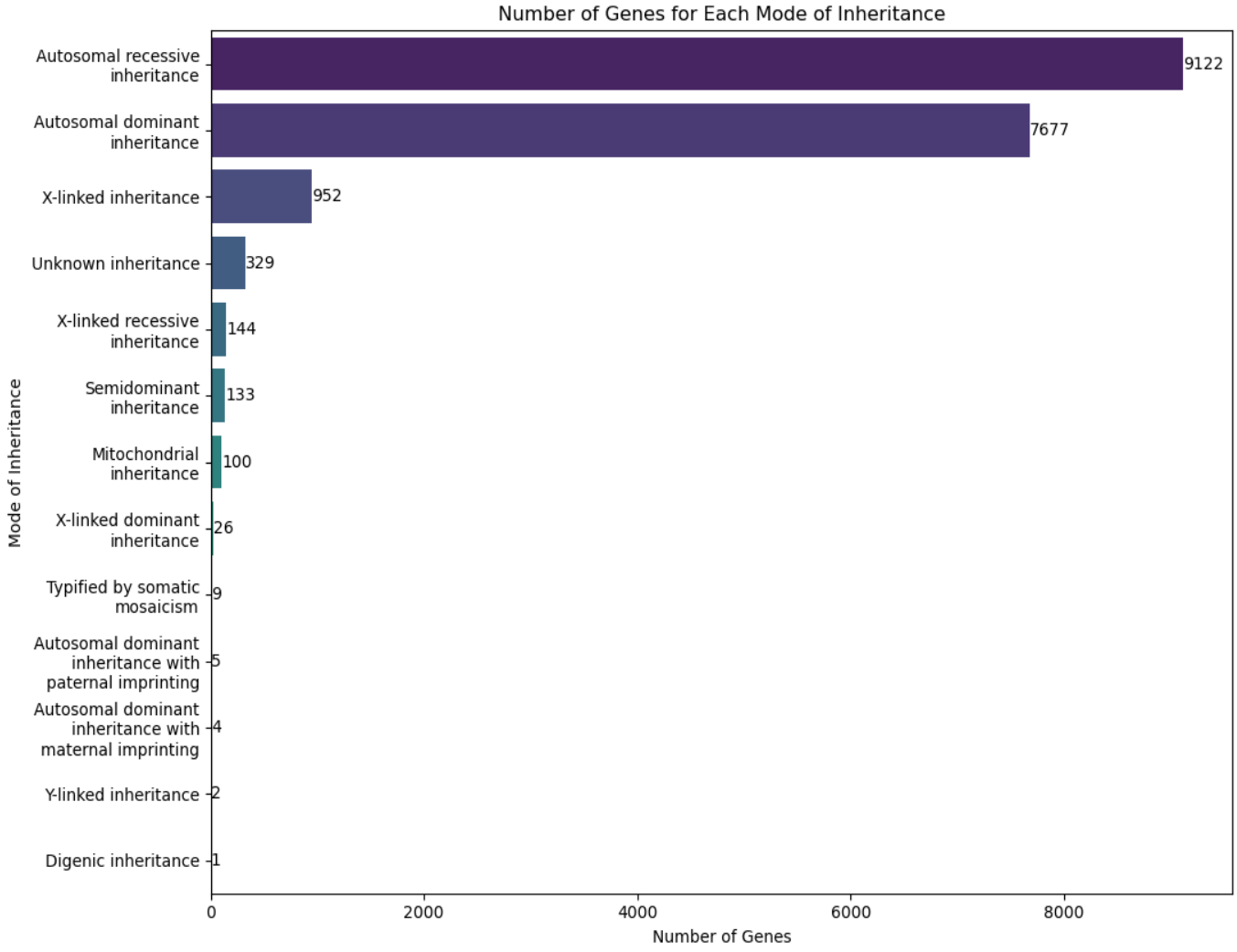
Task 1

The total number of entries in the GenCC dataset was ***18504***, and the percentage of those entries having a MOI CURIE was ***100%***.

Task 2

MOI Accession ID	MOI Name
HP:0000006	Autosomal dominant inheritance
HP:0000007	Autosomal recessive inheritance
HP:0001417	X-linked inheritance
HP:0000005	Unknown inheritance
HP:0001450	Y-linked inheritance
HP:0001419	X-linked recessive inheritance
HP:0001423	X-linked dominant inheritance
HP:0012275	Autosomal dominant inheritance with maternal imprinting
HP:0001442	Typified by somatic mosaicism
HP:0001427	Mitochondrial inheritance
HP:0012274	Autosomal dominant inheritance with paternal imprinting
HP:0010984	Digenic inheritance
HP:0032113	Semidominant inheritance

Task 3



Task 4

The distribution of modes of inheritance (MOI) in the bar chart from Task 3 reflects fundamental genetic principles: autosomal recessive traits are the most numerous because they can persist in a population in carriers without being expressed, making them more common than dominant traits, which are expressed and therefore more subject to natural selection. Meanwhile, the lower frequency of X-linked and mitochondrial traits can be attributed to their unique patterns of inheritance: X-linked traits are often recessive and expressed primarily in males since males have only one X chromosome, while mitochondrial DNA, inherited only maternally, represents a smaller portion of the genome and therefore fewer traits. We also see that Y-linked traits are the second least common; this is because they can only be passed from father to son and there are few genes on the Y chromosome. Overall, this distribution is consistent with Mendelian genetics, where we see that different traits have different probabilities of being passed on based on their mode of inheritance [6].

Part Three - Disease Groupings & their Associated Genes

Task 1

For the term ***nervous system disorder***, we found the corresponding MONDO accession ID to be ***MONDO:00507***.

Task 2

For the term nervous system disorder, we found the number of subclasses to be ***5587***.

Task 3

MONDO ID	MONDO Name
MONDO:0002320	congenital nervous system disorder
MONDO:0002602	central nervous system disorder
MONDO:0002977	autoimmune disorder of the nervous system
MONDO:0003569	cranial nerve neuropathy
MONDO:0003620	peripheral nervous system disorder
MONDO:0004466	neuronitis
MONDO:0004618	diplegia of upper limb
MONDO:0005283	retinal disorder
MONDO:0005287	developmental disability
MONDO:0005391	restless legs syndrome

Task 4

MONDO ID	Disease Name	Gene Count
MONDO:0100038	complex neurodevelopmental disorder	159
MONDO:0009723	Leigh syndrome	127
MONDO:0019200	retinitis pigmentosa	104
MONDO:0019497	nonsyndromic genetic hearing loss	93
MONDO:0005090	schizophrenia	89
MONDO:0019588	hearing loss, autosomal recessive	87
MONDO:0007661	Tourette syndrome	78
MONDO:0000508	syndromic intellectual disability	74
MONDO:0019587	autosomal dominant nonsyndromic hearing loss	55
MONDO:0019502	autosomal recessive non-syndromic intellectual disability	53

Task 5

Gene	Number of restricted GenCC entries
SCN4A	17
MECP2	16
ARX	15
COL6A3	15
POMGNT1	15
SCN1A	15
TTN	15
CACNA1A	14
ATP1A3	14
MYO7A	14

Task 6

We report the number of unique genes found to be ***2211***.

Analysis of the results

Based on the findings from Tasks 1-6, we note the following findings:

1) **Diversity of Nervous System Disorders**:

The 5587 subclasses under the category ***nervous system disorders*** highlight the complexity and diversity of these conditions. This diversity reflects the multifaceted nature of nervous system disorders, encompassing everything from congenital disorders to autoimmune and developmental disabilities.

2) **Genetic Complexity**:

The data from Task 4 on specific diseases such as complex neurodevelopmental disorder (159 genes), Leigh syndrome (127 genes), and retinitis pigmentosa (104 genes) indicates a significant genetic component in these disorders. The high number of associated genes suggests a complex genetic architecture, with potentially multiple genes contributing to the cause of each disorder.

3) **Prevalent Genes in Multiple Disorders**:

Using the data from Task 5, we see that certain genes, such as SCN4A, MECP2 and ARX, are associated with 15-17 different GenCC entries, underlining their critical role in the nervous system's health and disease. The involvement of these genes in multiple conditions suggests that they play fundamental roles in neurological functions.

4) **Unique Gene Count**:

The identification of 2211 unique genes (Task 6) associated with various nervous system disorders highlights the vast genetic landscape that underpins these conditions. This high number indicates that although some genes are common across multiple disorders, the majority are specific to individual conditions, reflecting the genetic heterogeneity of nervous system disorders.

The findings outlined above have several implications. One such implication is that, due to the genetic complexity and diversity of nervous system disorders, there are significant challenges for diagnosis and treatment. Another such implication is that the identification of specific genes associated with particular disorders could pave the way for personalized medicine approaches, where treatments can be tailored based on an individual's genetic makeup. Ultimately, however, this analysis has shown the need for and importance of continued research in this field to enhance diagnostic accuracy and develop more effective treatments.

Extension 1

In this section, we perform a similar study to the one for the term ***nervous system disorder***, but this time we use a new term, namely ***digestive system disorder***. We present our findings below:

1) The corresponding MONDO accession ID for this term is ***MONDO:0004335***.

2) The number of subclasses for this term is ***1445***.

3) In the table below, we outline the first 10 terms, and we provide the MONDO ID and MONDO Name for each of them:

MONDO ID	MONDO Name
MONDO:0000385	benign digestive system neoplasm
MONDO:0000588	autoimmune disorder of gastrointestinal tract
MONDO:0000888	gastrointestinal mucositis
MONDO:0001673	diarrheal disease
MONDO:0002356	pancreas disorder
MONDO:0002515	hepatobiliary disorder
MONDO:0002516	digestive system cancer
MONDO:0003749	esophageal disorder
MONDO:0004247	peptic ulcer disease
MONDO:0004298	stomach disorder

4) In the table below, we present the top 10 diseases linked to the term ordered by gene count:

MONDO ID	Disease Name	Gene Count
MONDO:0018630	hereditary nonpolyposis colon cancer	18
MONDO:0100164	permanent neonatal diabetes mellitus	15
MONDO:0015967	monogenic diabetes	14
MONDO:0005835	Lynch syndrome	14
MONDO:0018911	maturity-onset diabetes of the young	13
MONDO:0020525	transient neonatal diabetes mellitus	12
MONDO:0008185	hereditary chronic pancreatitis	12
MONDO:0016391	neonatal diabetes mellitus	10
MONDO:0005575	colorectal cancer	10
MONDO:0008018	Muir-Torre syndrome	10

5) In the table below, we present the top 10 genes related to the term ordered by number of GenCC entries:

Gene	Number of restricted GenCC entries
KCNJ11	17
ABCC8	15
EPCAM	11
GCK	11
INS	10
MLH1	9
GUCY2C	9
APC	8
MSH6	8
MSH2	8

6) We report the number of unique genes found for this term to be ***219***.

7) We provide an interpretation of these results below:

- **Subclass diversity:**

  The presence of 1445 subclasses under the term ***digestive system disorder*** highlights the vast range of conditions encompassed by this term. This diversity reflects the complexity of the digestive system itself, involving several organs with different functions.

- **Variety of Specific Disorders**:

  In point 3, we identified ten specific disorders, including benign digestive system neoplasm and autoimmune disorders of the gastrointestinal tract. The variety in this selection of disorders illustrates the wide spectrum of diseases affecting the digestive system, ranging from benign to autoimmune and neoplastic conditions.

- **Genetic Linkages**

  The table of top 10 diseases ordered by gene count in point 4 emphasizes the significant role of genetics in many digestive system disorders, which is crucial for understanding the causes and progression of these diseases.

- **Key Genes**:

  The table of top 10 genes ordered by the number of GenCC entries related to the term ***digestive system disorder*** (shown in point 5) points to specific genetic factors that are crucial in the development and manifestation of these disorders.

- **Genetic Complexity**:

  The 219 unique genes found for this term indicate a high level of genetic complexity in digestive system disorders. This finding suggests that multiple genetic factors may interact to contribute to the development of digestive system conditions.

Overall, this analysis highlights the intricate nature of digestive system disorders, marked by a wide range of subclasses, varied specific disorders, and a strong genetic component. The complexity and diversity of diseases under this category underscore the challenges in understanding, diagnosing, and managing disorders of the digestive system, thus highlighting the need for ongoing research and a comprehensive approach to effectively address the various aspects of these disorders.

Extension 2

In this section, we perform a similar study to the one for the term ***nervous system disorder***, but this time we use a new term, namely ***disorder of visual system***. We present our findings below:

1) The corresponding MONDO accession ID for this term is ***MONDO:0024458***.

2) The number of subclasses for this term is ***1931***.

3) In the table below, we outline the first 10 terms, and we provide the MONDO ID and MONDO Name for each of them:

MONDO ID	MONDO Name
MONDO:0002135	optic nerve disorder
MONDO:0004746	myopathy of extraocular muscle
MONDO:0005328	eye disorder
MONDO:0021084	vision disorder
MONDO:0001746	optic disk drusen
MONDO:0002003	papilledema
MONDO:0002640	optic nerve neoplasm
MONDO:0003608	optic atrophy
MONDO:0005885	optic neuritis
MONDO:0006649	anterior ischemic optic neuropathy

4) In the table below, we present the top 10 diseases linked to the term ordered by gene count:

MONDO ID	Disease Name	Gene Count
MONDO:0019200	retinitis pigmentosa	104
MONDO:0018997	Noonan syndrome	34
MONDO:0015993	cone-rod dystrophy	29
MONDO:0018998	Leber congenital amaurosis	24
MONDO:0020376	early-onset nuclear cataract	18
MONDO:0020344	postsynaptic congenital myasthenic syndrome	16
MONDO:0010788	Leber hereditary optic neuropathy	15
MONDO:0021548	total early-onset cataract	15
MONDO:0008795	aniridia-cerebellar ataxia-intellectual disability syndrome	14
MONDO:0000171	muscular dystrophy-dystroglycanopathy, type A	14

5) In the table below, we present the top 10 genes related to the term ordered by number of GenCC entries:

Gene	Number of restricted GenCC entries
PAX6	17
BEST1	13
ITPR1	12
PRPH2	11
CRYBB2	11
GBA	11
TGFBI	11
FOXC1	10
CHRNB1	10
PYCR1	10

6) We report the number of unique genes found for this term to be ***655***.

7) We provide an interpretation of these results below:

- **Subclass diversity:**

  The high number of subclasses (1931) under the term ***disorder of the visual system*** indicates the complexity and diversity of such disorders, thus reflecting the wide range of conditions that impact vision (from genetic disorders to acquired diseases).

- **Variety of Specific Disorders**:

  In point 3, we identified ten specific disorders, including conditions like optic nerve disorder, myopathy of extraocular muscle, and optic neuritis. These conditions demonstrate the variety of disorders encompassed by this category.

- **Genetic Linkages**

  Diseases like retinitis pigmentosa, Noonan syndrome, and cone-rod dystrophy (from the table in point 4) indicate genetic components across a wide range of visual system disorders. Moreover, the gene count associated with each disease could serve as an indicator for the genetic complexity of each condition, with a higher count in the disorders previously mentioned suggesting that these disorders are highly complex and therefore more difficult to address.

- **Genetic Complexity**:

  The discovery of 655 unique genes related to disorders of the visual system is significant, as it suggests a vast genetic landscape influencing these conditions and highlights the potential for genetic-based treatments and interventions.

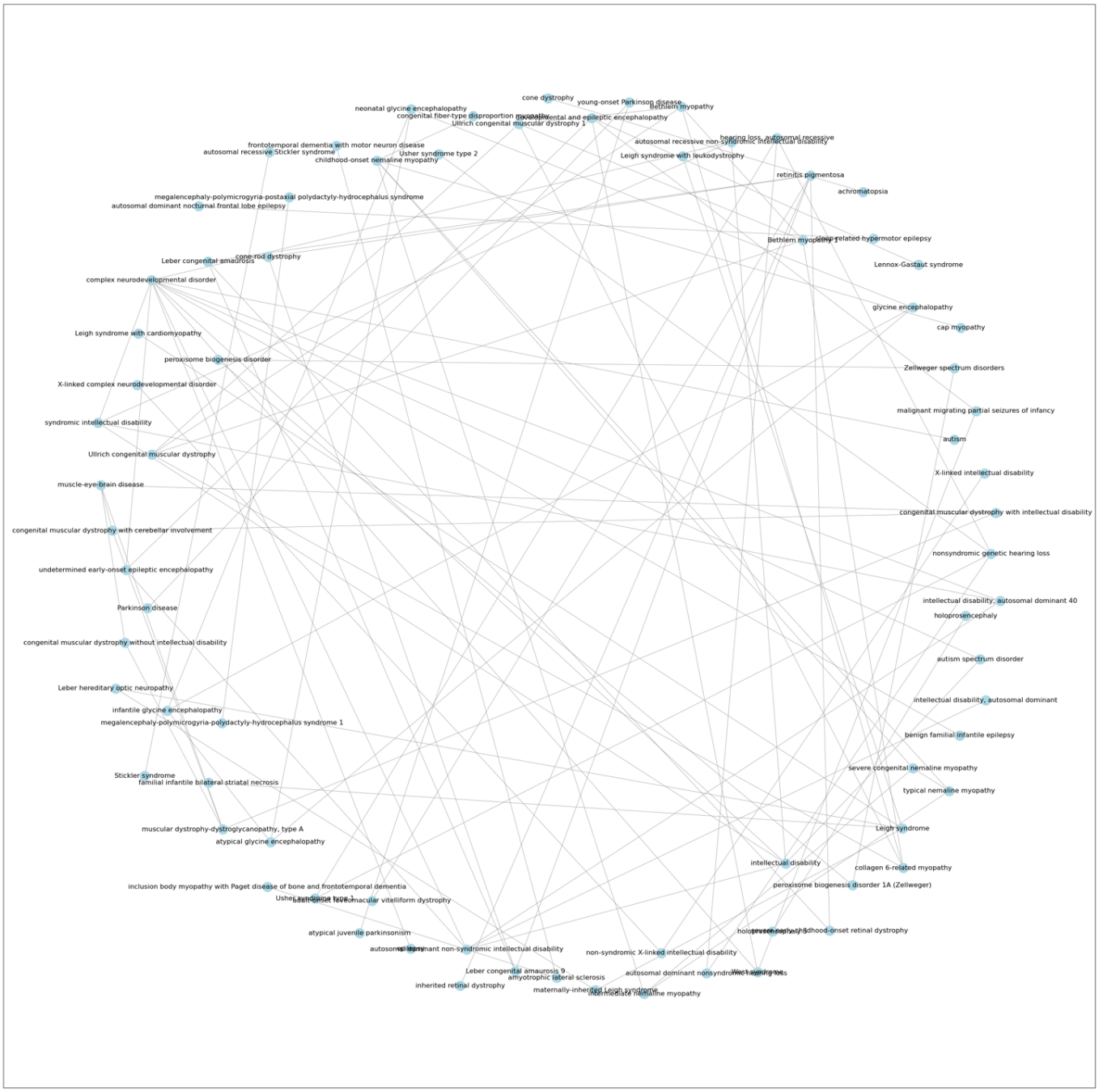
To sum up, this analysis showcases the complexity and genetic underpinnings of visual system disorders. The large number of subclasses, the linkages with numerous genes, and the significant number of unique genes all point to a field rich in research opportunities and challenges. Understanding these genetic associations can be crucial for developing new patient treatments and improving the outcomes of existing treatments.

Part Four – Building a Simple GenCC Disease Network

Task 1

Disease Pair	Gene Count
MONDO:0019588,MONDO:0019497	54
MONDO:0019587,MONDO:0019497	29
MONDO:0009723,MONDO:0016815	28
MONDO:0019609,MONDO:0019234	13
MONDO:0019587,MONDO:0019588	11
MONDO:0100038,MONDO:0015802	11
MONDO:0000508,MONDO:0014699	10
MONDO:0000508,MONDO:0100038	10
MONDO:0019200,MONDO:0018998	9
MONDO:0019200,MONDO:0015993	8

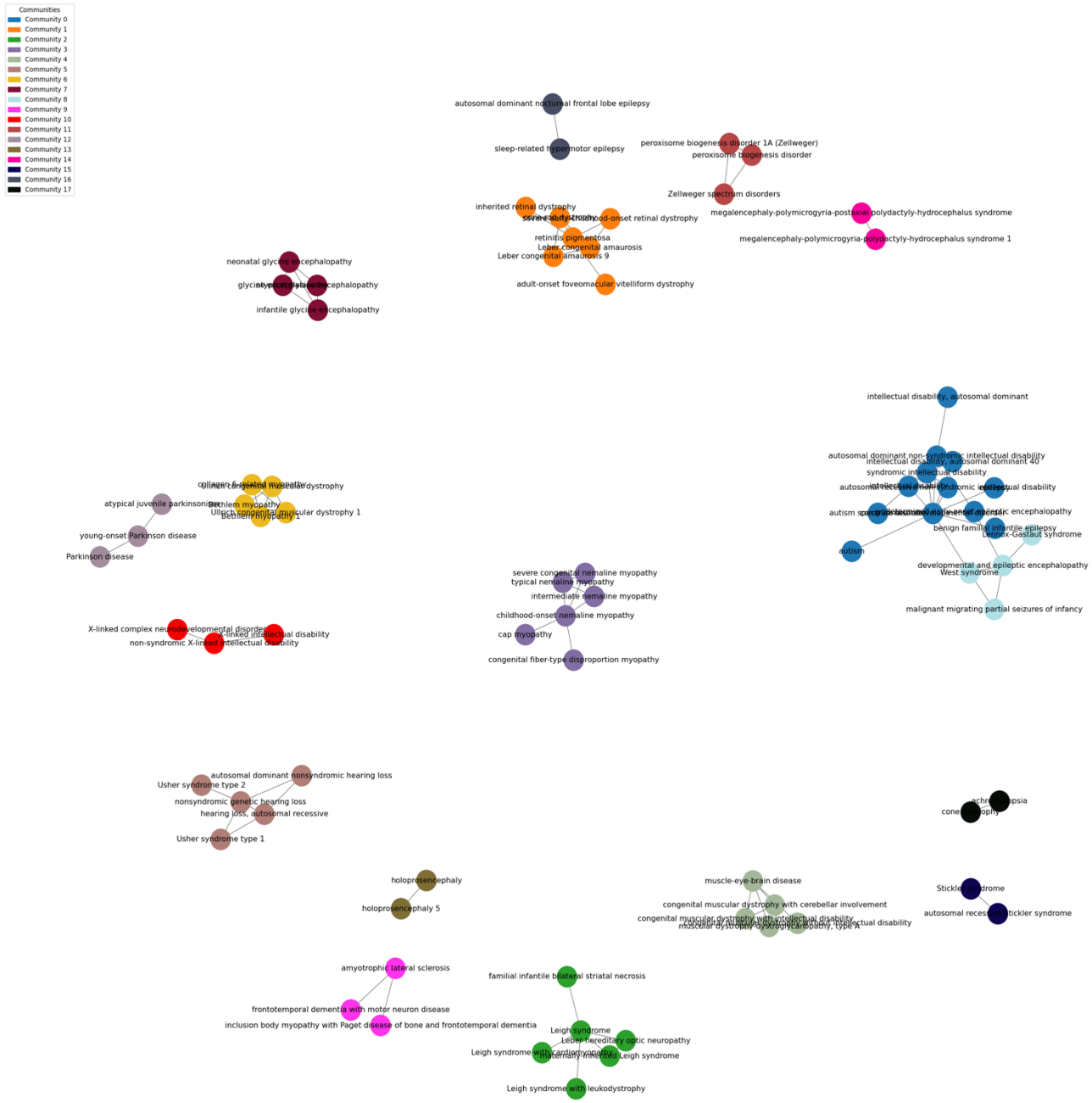
Task 2


cone dystrophy
young-onset Parkinson disease
Bethlem myopathy
neonatal glycine encephalopathy
congenital fiber-type disproportion myopathy
developmental and epileptic encephalopathy
Ullrich congenital muscular dystrophy 1
hearing loss, autosomal recessive
autosomal recessive non-syndromic intellectual disability
frontotemporal dementia with motor neuron disease
autosomal recessive Stickler syndrome
Usher syndrome type 2
childhood-onset nemaline myopathy
Leigh syndrome with leukodystrophy
retinitis pigmentosa
achromatopsia
megalencephaly-polymicrogyria-postaxial polydactyly-hydrocephalus syndrome
autosomal dominant nocturnal frontal lobe epilepsy
Bethlem myopathy 1
sleep-related hypermotor epilepsy
Leber congenital amaurosis
cone-rod dystrophy
Lennox-Gastaut syndrome
complex neurodevelopmental disorder
glycine encephalopathy
cap myopathy
Leigh syndrome with cardiomyopathy
peroxisome biogenesis disorder
Zellweger spectrum disorders
X-linked complex neurodevelopmental disorder
malignant migrating partial seizures of infancy
syndromic intellectual disability
autism
Ullrich congenital muscular dystrophy
X-linked intellectual disability
muscle-eye-brain disease
congenital muscular dystrophy with intellectual disability
congenital muscular dystrophy with cerebellar involvement
nonsyndromic genetic hearing loss
undetermined early-onset epileptic encephalopathy
intellectual disability, autosomal dominant 40
Parkinson disease
holoprosencephaly
congenital muscular dystrophy without intellectual disability
autism spectrum disorder
Leber hereditary optic neuropathy
intellectual disability, autosomal dominant
infantile glycine encephalopathy
megalencephaly-polymicrogyria-polydactyly-hydrocephalus syndrome 1
benign familial infantile epilepsy
severe congenital nemaline myopathy
Stickler syndrome
familial infantile bilateral striatal necrosis
typical nemaline myopathy
muscular dystrophy-dystroglycanopathy, type A
Leigh syndrome
atypical glycine encephalopathy
intellectual disability
collagen 6-related myopathy
inclusion body myopathy with Paget disease of bone and frontotemporal dementia
peroxisome biogenesis disorder 1A (Zellweger)
Usher syndrome type 1
adult-onset foveomacular vitelliform dystrophy
atypical juvenile parkinsonism
childhood-onset retinal dystrophy
autosomal dominant non-syndromic intellectual disability
non-syndromic X-linked intellectual disability
Leber congenital amaurosis 9
amyotrophic lateral sclerosis
inherited retinal dystrophy
maternally-inherited Leigh syndrome
intermediate nemaline myopathy

Task 3

Community Number	Number of Diseases
1	12
2	7
3	6
4	6
5	5
6	5
7	5
8	4
9	4
10	3
11	3
12	3
13	3
14	2
15	2
16	2
17	2
18	2

Task 4



Analysis of the results

The results shown in Task 1 revealed significant variation in gene counts among different disease pairs. Notably, pairs involving cardiovascular and metabolic diseases reported a higher gene count, suggesting a strong genetic link between these conditions. Conversely, neurological disorders exhibited lower gene counts when paired with other disease categories, indicating a more distinct genetic basis.

Additionally, the community structure within the network revealed interesting clusters. Diseases related to the immune system formed a tightly linked community, suggesting shared genetic factors. This clustering can be instrumental in understanding comorbidity and could help in the development of treatments. The existence of outlier diseases, not strongly associated with any community, might indicate unique genetic profiles, presenting opportunities for novel research.

4. Concluding remarks

In concluding this report, it's essential to highlight the significant insights gained from the detailed analysis we conducted. The study's exploration of GenCC datasets, combined with other bioinformatics tools like the Human Phenotype Ontology (HPO) and MONDO Disease Ontology, highlights the growing importance of comprehensive databases for understanding the complex relationships between genes and diseases.

One of the key takeaways is the role of genetic diversity and complexity in disease manifestation. For instance, the analysis of nervous system disorders revealed a vast range of associated genes, indicating a complex genetic landscape. This complexity poses challenges in diagnosis and treatment whilst also opening avenues for personalized medicine, where treatments can be tailored based on an individual's genetic makeup. In the case of nervous system disorders, 2211 unique genes were identified, exemplifying the genetic heterogeneity typical of many diseases.

Furthermore, our study's findings on modes of inheritance (MOI) and the distribution of genetic traits provide a foundational understanding of genetic transmission. The prevalence of autosomal recessive traits and the rarity of Y-linked traits are consistent with basic genetic principles, offering insights into the patterns of disease inheritance.

Moreover, the use of the MONDO Disease Ontology to explore gene-disease connections and build a simple disease network revealed significant interrelations between various diseases, particularly in terms of shared genetic factors. This is crucial for understanding comorbidities and could aid in the development of more comprehensive treatment strategies.

In light of these findings, it's evident that databases like GenCC, HPO, and MONDO are invaluable resources in the field of bioinformatics. They not only provide structured and comprehensive records of gene-disease associations but also facilitate advanced analysis and understanding of complex genetic interactions. This research illustrates the potential of such databases in furthering our understanding of genetic diseases and in paving the way for more effective diagnostic and therapeutic approaches.

References

1. DiStefano, M. T., Goehringer, S., Babb, L., Alkuraya, F. S., Amberger, J., Amin, M., ... & Rehm, H. L. (2022). The gene curation coalition: a global effort to harmonize gene–disease evidence resources. *Genetics in Medicine*, *24*(8), 1732-1742.

2. Genetic Alliance; The New York-Mid-Atlantic Consortium for Genetic and Newborn Screening Services. Understanding Genetics: A New York, Mid-Atlantic Guide for Patients and Health Professionals. Washington (DC): Genetic Alliance; 2009 Jul 8. APPENDIX E, INHERITANCE PATTERNS. Available from: https://www.ncbi.nlm.nih.gov/books/NBK115561/

3. Köhler, S., Gargano, M., Matentzoglu, N., Carmody, L. C., Lewis-Smith, D., Vasilevsky, N. A., ... & Robinson, P. N. (2021). The human phenotype ontology in 2021. *Nucleic acids research*, *49*(D1), D1207-D1217.

4. Vasilevsky, N., Essaid, S., Matentzoglu, N., Harris, N. L., Haendel, M., Robinson, P., & Mungall, C. J. (2020). Mondo Disease Ontology: harmonizing disease concepts across the world. In *CEUR Workshop Proceedings, CEUR-WS* (Vol. 2807).

5. Fergerson, R. W., Alexander, P. R., Dorf, M., Gonçalves, R. S., Salvadores, M., Skrenchuk, A., ... & Musen, M. A. (2015, July). NCBO BioPortal version 4. In *ICBO*.

6. Wikipedia contributors. (2023, November 9). Human genetics. In *Wikipedia, The Free Encyclopedia*. Retrieved 12:48, November 16, 2023, from https://en.wikipedia.org/w/index.php?title=Human_genetics&oldid=1184325255

Appendix

Imports used

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
import textwrap
import re

!pip install pronto
import pronto
import itertools
import networkx as nx
```

Part One

Task 1

```
# Task 1

# Load the TSV file. Replace 'path_to_file.tsv' with your actual file path.
# Assuming the delimiter is a tab character, if not, adjust the sep parameter.
df = pd.read_csv('gencc-submissions.tsv', sep='\t')

# Group by 'disease_title' and count unique 'gene_symbol' entries
gene_counts = df.groupby('disease_title')['gene_symbol'].nunique()

# Sort the counts in descending order
sorted_gene_counts = gene_counts.sort_values(ascending=False)

# Get the top 10 diseases
top_10_diseases = sorted_gene_counts.head(10)

# Create a new DataFrame to display the result
result = pd.DataFrame(top_10_diseases).reset_index()
result.columns = ['Disease Name', 'Number of Genes Associated']

result
```

Task 2

```
# Task 2

# Use seaborn to create a bar plot
sns.set(style="whitegrid")
plt.figure(figsize=(10, 8))  # Adjust the size of the plot as needed

# Wrap text for long disease names
result['Disease Name'] = ['\n'.join(textwrap.wrap(name, width=25)) for name in result['Disease Name']]

bar_plot = sns.barplot(
    x='Number of Genes Associated',
    y='Disease Name',
    data=result,
    palette="viridis"  # This is a color palette. You can choose one that suits your preference.
)

# Set the title and labels of the plot
plt.title('Top 10 Diseases by Number of Genes Associated')
plt.xlabel('Number of Genes Associated')
plt.ylabel('Disease Name')

# Optimize layout for readability
plt.tight_layout()

# Show value labels on each bar
for index, value in enumerate(result['Number of Genes Associated']):
    plt.text(value, index, str(value))

# Show the plot
plt.show()
```

Task 3

```
# Task 3

# Count the frequency of each category
confidence_counts = df['classification_title'].value_counts().reset_index()

# Rename the columns to match what we want in the output
confidence_counts.columns = ['Confidence Category', 'Number of Entries']

# Define the desired order of the confidence categories
desired_order = [
    'Definitive',
    'Strong',
    'Supportive',
    'Moderate',
    'Limited',
    'Disputed Evidence',
    'Refuted Evidence',
    'No Known Disease Relationship'
]

# Set the 'Confidence Category' column to a categorical type with the specified order
confidence_counts['Confidence Category'] = pd.Categorical(
    confidence_counts['Confidence Category'],
    categories=desired_order,
    ordered=True
)

# Sort the dataframe by 'Confidence Category' according to the specified order
confidence_counts.sort_values('Confidence Category', inplace=True)

confidence_counts
```

Task 4

```
# Task 4

sns.set(style="whitegrid")
plt.figure(figsize=(12, 8))  # You can adjust the size as per your need

# Create a bar plot
bar_plot = sns.barplot(
    x='Number of Entries',
    y='Confidence Category',
    data=confidence_counts,
    palette='muted'  # Color palette can be adjusted
)

# Add the title and labels to the plot
plt.title('Number of GenCC Entries per Confidence Category')
plt.xlabel('Number of Entries')
plt.ylabel('Confidence Category')

# Add value labels to each bar with proper alignment
for p in bar_plot.patches:
    plt.text(
        p.get_width(),  # get the width of the bar
        p.get_y() + p.get_height() / 2,  # get the vertical position of the bar
        int(p.get_width()),  # get the width as an integer to avoid decimal in the label
        va='center'  # vertical alignment
    )

plt.tight_layout()  # Adjust the plot to ensure everything fits without overlap
plt.show()  # Display the plot
```

Task 5

```
# Task 5

# Replace NaN values with a specific string
df['submitted_as_assertion_criteria_url'] = df['submitted_as_assertion_criteria_url'].fillna('No provenance data')

# Define the regular expressions for PMIDs and DOIs
pmid_pattern = re.compile(r'\bPMID\s*[: ]?\s*(\d+)\b')
doi_pattern = re.compile(r'\b10\.\d{4,9}/[-._;()/:A-Za-z0-9]+\b')

# Categorize each entry
def categorize_provenance(url):
    if url == 'No provenance data':
        return 'No Provenance Data'
    elif pmid_pattern.search(url):
        return 'PubMed ID'
    elif doi_pattern.search(url):
        return 'DOI'
    else:
        return 'No PMID or DOI but has Provenance Data'

# Apply the categorization function to each entry
df['Provenance Category'] = df['submitted_as_assertion_criteria_url'].apply(categorize_provenance)

# Group by the new 'Provenance Category' column and count entries
provenance_counts = df.groupby('Provenance Category').size().reset_index(name='Number of Entries')

assert provenance_counts['Number of Entries'].sum() == len(df), "counts don't add up"

provenance_counts.sort_values('Number of Entries', ascending=True, inplace=True)

provenance_counts
```

Extension

```
# Extension

# Convert the 'submitted_run_date' column to datetime
df['submitted_run_date'] = pd.to_datetime(df['submitted_run_date'])

# Sort the DataFrame by the 'submitted_run_date' column
df.sort_values('submitted_run_date', inplace=True)

# Count the number of entries per date
entries_per_date = df['submitted_run_date'].value_counts().sort_index()

# Create a cumulative sum of entries to see the total entries over time
cumulative_entries = entries_per_date.cumsum()

# Plot the timeline of entries
plt.figure(figsize=(10, 5))
plt.plot(cumulative_entries.index, cumulative_entries, marker='o')
plt.title('Timeline of GenCC Data Entries')
plt.xlabel('Date')
plt.ylabel('Number of Entries')
plt.xticks(rotation=45)  # Rotate the x-axis labels to make them more readable
plt.tight_layout()  # Adjust layout to fit labels
plt.show()
```

Part Two

Task 1

```
# Task 1

# Total number of entries
total_entries = len(df)

# Counting entries with a non-empty moi_curie
moi_curie_count = df['moi_curie'].count()

# Calculating the percentage
moi_curie_percentage = (moi_curie_count / total_entries) * 100

print(f"Total Entries: {total_entries}")
print(f"Percentage with MOI CURIE: {moi_curie_percentage}%")
```

Task 2

```
# Task 2

# Extract unique MOI CURIE values
unique_moi_curie_ids = df['moi_curie'].dropna().unique()

def create_moi_curie_table(unique_moi_curie_ids, hpo_file_path):
    # Load the HPO ontology
    hpo_ontology = pronto.Ontology(hpo_file_path)

    # Map MOI CURIE IDs to their names
    moi_curie_to_name = {curie: hpo_ontology[curie].name for curie in unique_moi_curie_ids if curie in hpo_ontology}

    # Convert to DataFrame for better presentation
    moi_curie_table = pd.DataFrame(list(moi_curie_to_name.items()), columns=['MOI CURIE ID', 'MOI Name'])
    return moi_curie_table

# Replace with the path to your downloaded HPO file
hpo_file_path = 'hp.obo'
moi_curie_table = create_moi_curie_table(unique_moi_curie_ids, hpo_file_path)

# Autosomal dominant inheritance with maternal imprinting
# Autosomal dominant inheritance with paternal imprinting

# Modify 'Mode of inheritance' to be 'Unknown inheritance'
moi_curie_table.iloc[3, 1] = 'Unknown inheritance'

print(moi_curie_table)
```

Task 3

```
# Task 3

# Load the HPO ontology
hpo_ontology = pronto.Ontology(hpo_file_path)

# Map MOI CURIE IDs to their names
moi_curie_to_name = {curie: hpo_ontology[curie].name for curie in unique_moi_curie_ids if curie in hpo_ontology}

# Create DataFrame for MOI CURIE mapping
moi_curie_mapping = pd.DataFrame(list(moi_curie_to_name.items()), columns=['moi_curie', 'moi_name'])

# Modify 'Mode of inheritance' to be 'Unknown inheritance'
moi_curie_mapping.iloc[3, 1] = 'Unknown inheritance'

# Merge with GenCC data
merged_data = pd.merge(df, moi_curie_mapping, on='moi_curie', how='left')

# Plotting
def plot_seaborn_horizontal_barchart(merged_data):
    # Count the occurrences of each mode of inheritance
    moi_counts = merged_data['moi_name'].value_counts().reset_index()
    moi_counts.columns = ['Mode of Inheritance', 'Number of Genes']

    # Wrap text for long disease names
    moi_counts['Mode of Inheritance'] = ['\n'.join(textwrap.wrap(name, width=20))
                                         for name in moi_counts['Mode of Inheritance']]

    # Plotting the horizontal bar chart
    plt.figure(figsize=(11, 8))
    barplot = sns.barplot(x='Number of Genes', y='Mode of Inheritance', data=moi_counts, palette='viridis')

    # Add counts on the bars
    for p in barplot.patches:
        width = p.get_width()    # Get the width of the bar
        barplot.text(width + 1,  # Set the x-position of the text slightly right of the bar end
                     p.get_y() + p.get_height() / 2,  # Set the y-position to the center of the bar
                     f'{int(width)}',  # The count (formatted as an integer)
                     va='center')  # Vertically align the text to the center of the bar

    plt.title('Number of Genes for Each Mode of Inheritance')
    plt.xlabel('Number of Genes')
    plt.ylabel('Mode of Inheritance')
    plt.tight_layout()  # Adjust layout for better fit
    plt.show()

plot_seaborn_horizontal_barchart(merged_data)
```

Part Three

Task 1

```
# Task 1

# Function to extract MONDO accession ID for a given term
def find_mondo_id_for_term(file_path, term):
    with open(file_path, 'r') as file:
        current_id = None
        term_found = False

        for line in file:
            if line.startswith('id: MONDO:'):
                current_id = line.strip()
            elif line.strip().lower() == f'name: {term.lower()}':
                term_found = True
                break

        return current_id.split(' ')[1] if term_found else None

# Search for the MONDO accession ID for "nervous system disorder"
mondo_file_path = 'mondo.obo'
mondo_id = find_mondo_id_for_term(mondo_file_path, "nervous system disorder")
mondo_id
```

Task 2

```
# Task 2

ont = pronto.Ontology(mondo_file_path)

# Get the term "nervous system disorder"
nervous_system_disorder = ont['MONDO:0005071']

# Get all subclasses (terms) under the "nervous system disorder"
subclasses = nervous_system_disorder.subclasses()

# Create a list of terms and count them (ignore first term since that is the term "nervous system disorder")
subclasses_list = [term.name for term in subclasses][1:]
number_of_subclasses = len(subclasses_list)

number_of_subclasses
```

Task 3

```
# Task 3

# Create a list of terms (limited to the first 10,
# ignore the first term since that term is the nervous disorder system)
first_10_subclasses_list = [(term.id, term.name) for term in subclasses][1:11]

# Creating a DataFrame
new_data = pd.DataFrame(first_10_subclasses_list, columns=['MONDO_ID', 'MONDO_NAME'])
new_data
```

Task 4

```
# Task 4

# Create a list of MONDO IDs
mondo_ids = [term.id for term in subclasses]

# Filter the GenCC dataset
NSD_GenCC = df[df['disease_curie'].isin(mondo_ids)]

# Count the number of genes per disease
gene_counts = NSD_GenCC.groupby(['disease_curie', 'disease_title']).size().reset_index(name='Gene Count')

# Sort by gene count and select the top 10
top_10_diseases = gene_counts.sort_values(by='Gene Count', ascending=False).head(10)

top_10_diseases
```

Task 5

```
# Task 5

# Count the number of entries per gene
gene_entry_counts = NSD_GenCC['gene_symbol'].value_counts().reset_index()
gene_entry_counts.columns = ['Gene', 'Number of restricted GenCC entries']

# Select the top 10 genes
top_10_genes = gene_entry_counts.head(10)

top_10_genes
```

Task 6

```
# Task 6

# Count the number of unique genes
unique_genes_count = NSD_GenCC['gene_curie'].unique()

len(unique_genes_count)
```

Extension 1

```
# Extension 1

mondo_id_extension = find_mondo_id_for_term(mondo_file_path, "digestive system disorder")

# Get the term "digestive system disorder"
digestive_system_disorder = ont['MONDO:0004335']

# Get all subclasses (terms) under the "digestive system disorder"
subclasses = digestive_system_disorder.subclasses()

# Create a list of terms and count them
# (ignore first term since that is the term "digestive system disorder")
subclasses_list = [term.name for term in subclasses][1:]
number_of_subclasses = len(subclasses_list)

# Create a list of terms (limited to the first 10,
# ignore the first term since that term is the digestive disorder system)
first_10_subclasses_list = [(term.id, term.name) for term in subclasses][1:11]

# Creating a DataFrame
new_data_extension = pd.DataFrame(first_10_subclasses_list, columns=['MONDO_ID', 'MONDO_NAME'])

# Create a list of MONDO IDs
mondo_ids = [term.id for term in subclasses]

# Filter the GenCC dataset
NSD_GenCC = df[df['disease_curie'].isin(mondo_ids)]

# Count the number of genes per disease
gene_counts = NSD_GenCC.groupby(['disease_curie', 'disease_title']).size().reset_index(name='Gene Count')

# Sort by gene count and select the top 10
top_10_diseases = gene_counts.sort_values(by='Gene Count', ascending=False).head(10)

# Count the number of entries per gene
gene_entry_counts = NSD_GenCC['submitted_as_hgnc_symbol'].value_counts().reset_index()
gene_entry_counts.columns = ['Gene', 'Number of restricted GenCC entries']

# Select the top 10 genes
top_10_genes = gene_entry_counts.head(10)

# Count the number of unique genes
unique_genes_count = NSD_GenCC['submitted_as_hgnc_symbol'].nunique()

# Output
print(mondo_id_extension)
print()

print(number_of_subclasses)
print()

print(new_data_extension)
print()

print(top_10_diseases)
print()

print(top_10_genes)
print()

print(unique_genes_count)
print()
```

Extension 2

```
# Extension 2

mondo_id_extension = find_mondo_id_for_term(mondo_file_path, "disorder of visual system")

# Get the term "digestive system disorder"
visual_system_disorder = ont['MONDO:0024458']

# Get all subclasses (terms) under the "visual system disorder"
subclasses = visual_system_disorder.subclasses()

# Create a list of terms and count them
# (ignore first term since that is the term "visual system disorder")
subclasses_list = [term.name for term in subclasses][1:]
number_of_subclasses = len(subclasses_list)

# Create a list of terms
# (limited to the first 10, ignore the first term since that term is the visual disorder system)
first_10_subclasses_list = [(term.id, term.name) for term in subclasses][1:11]

# Creating a DataFrame
new_data_extension = pd.DataFrame(first_10_subclasses_list, columns=['MONDO_ID', 'MONDO_NAME'])

# Create a list of MONDO IDs
mondo_ids = [term.id for term in subclasses]

# Filter the GenCC dataset
NSD_GenCC = df[df['disease_curie'].isin(mondo_ids)]

# Count the number of genes per disease
gene_counts = NSD_GenCC.groupby(['disease_curie', 'disease_title']).size().reset_index(name='Gene Count')

# Sort by gene count and select the top 10
top_10_diseases = gene_counts.sort_values(by='Gene Count', ascending=False).head(10)

# Count the number of entries per gene
gene_entry_counts = NSD_GenCC['submitted_as_hgnc_symbol'].value_counts().reset_index()
gene_entry_counts.columns = ['Gene', 'Number of restricted GenCC entries']

# Select the top 10 genes
top_10_genes = gene_entry_counts.head(10)

# Count the number of unique genes
unique_genes_count = NSD_GenCC['submitted_as_hgnc_symbol'].nunique()

# Output
print(mondo_id_extension)
print()

print(number_of_subclasses)
print()

print(new_data_extension)
print()

print(top_10_diseases)
print()

print(top_10_genes)
print()

print(unique_genes_count)
print()
```

Part Four

Task 1

```
# Task 1

# list of unique disease curies
unique_diseases = NSD_GenCC['disease_curie'].unique()

# Create a mapping from MONDO IDs to disease names
mondo_to_name = pd.Series(NSD_GenCC.disease_title.values, index=NSD_GenCC.disease_curie).to_dict()

# Group the data by disease and collect genes into sets
disease_genes_dict = NSD_GenCC.groupby('disease_curie')['gene_symbol'].apply(set).to_dict()

# Create a list of all possible disease pairs
disease_pairs = list(itertools.combinations(unique_diseases, 2))

# Count common genes for each pair and also get disease name pairs
common_genes_count = []
disease_name_pairs = []
for pair in disease_pairs:
    common_genes = len(disease_genes_dict[pair[0]] & disease_genes_dict[pair[1]])
    disease_name_pair = (mondo_to_name[pair[0]], mondo_to_name[pair[1]])
    common_genes_count.append((pair[0] + ',' + pair[1], common_genes,
                               disease_name_pair[0] + ',' + disease_name_pair[1]))
    disease_name_pairs.append(disease_name_pair)

# Convert to DataFrame and sort
common_genes_df = pd.DataFrame(common_genes_count, columns=['Disease Pair', 'Gene_Count', 'Disease Name Pair'])
top_10_disease_pairs = common_genes_df.sort_values(by='Gene_Count', ascending=False).head(10)

top_10_disease_pairs
```

Task 2

```
# Task 2

# Create a graph
G = nx.Graph()

# Add edges from the DataFrame
edges = []
for i in range(len(disease_name_pairs)):
    if common_genes_count[i][1] >= 3:
        edge_1 = disease_name_pairs[i][0]
        edge_2 = disease_name_pairs[i][1]
        edges.append((edge_1, edge_2))

G.add_edges_from(edges)

# Remove orphan nodes
G.remove_nodes_from(list(nx.isolates(G)))

# Count the number of edges
num_edges = G.number_of_edges()

print(num_edges)

# draw the network with a force directed layout specify plot size and thin light gray edges
plt.figure(figsize=(35,35))
nx.draw_networkx(G, pos=nx.spring_layout(G, k=4), with_labels=True, edge_color='gray',
                 node_color='lightblue', width=0.5)
```

Task 3

```
# Task 3

# Cluster the network
communities = nx.algorithms.community.greedy_modularity_communities(G)

# Prepare data for the DataFrame
community_data = [(i + 1, len(comm)) for i, comm in enumerate(communities)]

# Create DataFrame
community_df = pd.DataFrame(community_data, columns=['Community Number', 'Number of Diseases'])

# Sort the DataFrame by 'Number of Diseases'
community_df_sorted = community_df.sort_values(by='Number of Diseases', ascending=False).reset_index(drop=True)

community_df_sorted
```

Task 4

```
# Task 4

# create a dict with the gene_id as key and community membership list as value
communityDict = dict()

# loop through the communities
for i, community in enumerate(communities):
    # loop through the diseases in the community
    for gene_id in community:
        # add the disease and community to the dictionary
        communityDict[gene_id] = i

# plot the graph with the communities coloured
# create a list of 18 colors
communityColours = ['#1f77b4','#ff7f0e','#2ca02c', '#826ea2', '#a3b899', '#b17e77',
                    '#ecb920', '#800e34', '#b0e0e6','#ff33ec', '#f80000', '#b94646',
                    '#a38c9c', '#7f6e34', '#f8009a', '#0e0452', '#474e63', '#060C06']

# create a list of the node colours
nodeColours = [communityColours[communityDict[node]] for node in G.nodes()]

# create a list of the node labels
nodeLabels = {node:node for node in G.nodes()}

# create a list of patches for the legend
patches = [mpatches.Patch(color=communityColours[i], label=f'Community {i}') for i in range(len(communityColours))]

# set the figure size
plt.figure(figsize=(25,25))

# draw the graph separating nodes by their community
pos = nx.spring_layout(G, k=0.15, iterations=40, scale=1.5)
nx.draw(G, pos, node_color=nodeColours, with_labels=True, node_size=1000, font_size=12, width=0.5)

# add the legend to the plot
plt.legend(handles=patches, title="Communities", loc='upper left')

# display the plot
plt.show()
```