Project Summary

1.Data Preparation

- **Data Groups**: Analysis involved four groups: GF, Nr1d1 (both WT and KO), Hdac3 (WT and KO), and Nfil3 (WT and KO).
- **Processing Steps**: The data preparation included aligning sequencing data, calculating FPKM values, and generating input files for Metacycle software.

  Original data: 4 groups:

  a. GF: GFRNA1.fastq, GFRNA2.fastq, GFRNA3.fastq, GFRNA4.fastq

  b. Nr1d1: - WT: ZK1.fastq, ZK2.fastq, ZK3.fastq, ZK4.fastq
  - KO: ZK5.fastq, ZK6.fastq, ZK7.fastq, ZK8.fastq

  c. Hdac3: - WT: ZK1.fastq, ZK2.fastq, ZK3.fastq, ZK4.fastq
  - KO: ZK5.fastq, ZK6.fastq, ZK7.fastq, ZK8.fastq

  d. Nfil3: - WT: ZK1.fastq, ZK2.fastq, ZK3.fastq, ZK4.fastq, ZK5.fastq, ZK6.fastq
  - KO: ZK7.fastq, ZK8.fastq, ZK9.fastq, ZK10.fastq, ZK11.fastq, ZK12.fastq

- **Python files:**

  User python file: ***WT_4_calfpkm.py***
  Get replication file: ***get_sec_fpkm.py***
  Paste together : nr1d1_WT.txt
  Take nr1d1 ZK1.fastq for example → MetaCylce input file Nr1d1_WT.txt

```
########################################################################
##########  group Nr1d1: ZK1.fastq -> get input file Nr1d1_WT.txt ############
########################################################################
``` linux
# ZK1 – ZK4
bowtie2-build mm39.fasta ref_genome

bowtie2 -x ref_genome -U ZK1.fastq -S ZK1.sam
samtools view -bS ZK1.sam > ZK1.bam
samtools sort ZK1.bam -o ZK1.sorted.bam
samtools index ZK1.sorted.bam
samtools flagstat ZK1.sorted.bam
featureCounts -a mm39.ncbiRefSeq.gtf -o counts_ZK1.txt ZK1.sorted.bam
```
``` python
Calculate fpkm: WT_4_calfpkm.py
Replicate: get_sec_fpkm.py
Paste ZK1_rep1 ZK1_rep2 – ZK4_rep1 ZK4_rep2 - > nr1d1_WT.txt
```

- **Input files for MetaCycle:**

  GF : GF.txt
  Nr1d1: nr1d1_WT.txt nr1d1_KO.txt
  Hdac3: hdac3_WT.txt hdac3_KO.txt
  Nfil3: nfil3_WT.txt nfil3_KO.txt

2.Run Metacycle software
```

The MetaCycle package is mainly used for detecting rhythmic signals from large scale time-series data. Depending on features of each time-series data, MetaCycle incorporates ARSER(ARS), JTK_CYCLE(JTK),and Lomb-Scargle(LS) properly for periodic signal detection, and it could also output integrated analysis results if required.

- **Purpose:** Metacycle was used to detect rhythmic signals in the time-series data.
- **Output:** This step produced multiple output files for each data group, but the focus was primarily on the meta2d results
  Reference : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5079475/
- **Run Metacycle**
  Take nr1d1_WT.txt for example

```
#########################################################################
 nr1d1_WT.txt (Metacycle software)-> get JTK_nr1d1_WT.txt, ARS_nr1d1_WT.txt, LS_nr1d1_WT.txt, meta2d_nr1d1_WT.txt
#########################################################################
```r
install.packages("`MetaCycle`")
#install.packages("tidyverse")
require(`MetaCycle`)
require(tidyverse)
meta2d(infile="nr1d1_WT.txt",filestyle="txt", outdir="example_nr1d1_WT",
  , timepoints=seq(0, 42, by=6),
  outIntegration="onlyIntegration"
```

- **MetaCyle result outputs:**Where we only cares about meta2d results:
  GF: meta2d_ GF.txt
  Nr1d1: meta2d_nr1d1_WT.txt, meta2d_nr1d1_KO.txt
  Hdac3: meta2d_hdac3_WT.txt, meta2d_hdac3_KO.txt
  Nfil3: meta2d_nfil3_WT.txt, meta2d_nfil3_KO.txt

3.Gene Significance Identification

- **Methodology:** Significant genes were identified using a Python script, focusing on genes with a meta2d p-value < 0.05 and FPKM > 1.
- **Results:** This step resulted in lists of significant genes for each data group.
- **Python file:** ***siggene.py***
  First, get initial significant genes: based on meta2d_pvalue <0.05 →genes_GF.txt
  Then, filter out the genes with less than 1 fpkm → GF_filtered_genelist.txt

```
```Linux
awk '$9 > 1 && $10 > 1 && $11 > 1 && $12 > 1 && $13 > 1 && $14 > 1 && $15 > 1 && $16 > 1' meta2d_GF.txt > GF_filtered_genelist.txt
```
```
```

Lastly, get final significant genes:Choose initial significant genes with more than 1 fpkm as final significant genes

```Linux
grep -xFf genes_GF.txt GF_filtered_genelist.txt >sig_GF.txt
```

- **Output :** sig_GF.txt, sig_nr1d1_WT.txt, sig_nr1d1_KO.txt, sig_hdac3_WT.txt, sig_hdac3_KO.txt, sig_nfil3_WT.txt, sig_nfil3_KO.txt

4.Define microbiota cycling genes

- **Finding Conventional Genes:** combine the significant genes from three different lists (sig_nr1d1_WT.txt, sig_hdac3_WT.txt, sig_nfil3_WT.txt) to find common genes across these conditions using Linux command-line tools. The result is a list of conventional genes stored in conventional_genes.txt.

```Linux
cat sig_nfil3_WT.txt sig_hdac3_WT.txt sig_nr1d1_WT.txt | sort | uniq > conventional_genes.txt
```

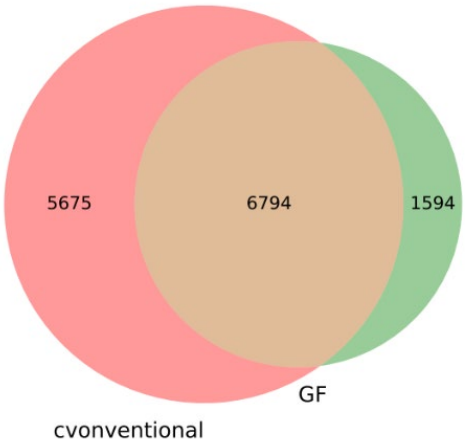
Data Integration: Combining and comparing gene lists from different conditions have provided a comprehensive view of gene significance relative to the presence or absence of gut microbiota (conventional vs. GF).

- **Finding Shared and Unique Genes:** refine gene lists to identify genes that are unique to WT (WT_only.txt), unique to GF (GF_only.txt), and shared between them (shared.txt). This step is crucial for dissecting the genetic differences and commonalities between conventional and germ-free environments.

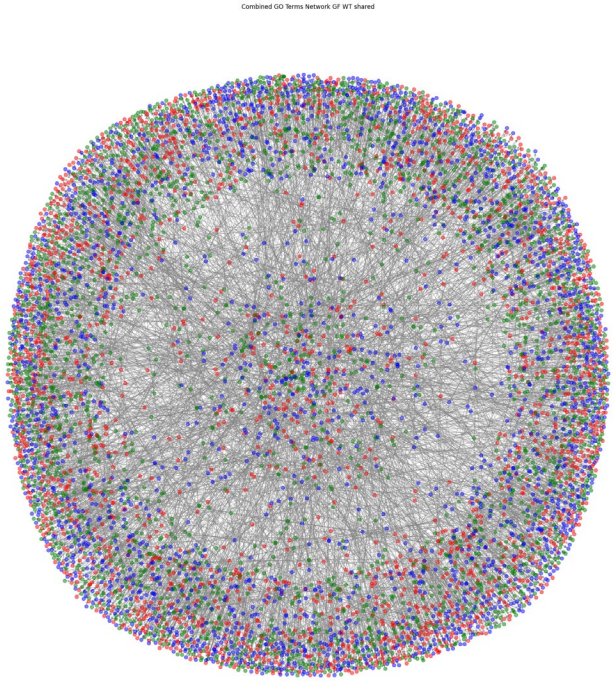
```Linux
sort sig_GF.txt > sig_GF_sorted.txt
sort conventional_genes.txt > conventional_genes_sorted.txt
comm -12 sig_GF_sorted.txt conventional_genes_sorted.txt > shared.txt
comm -23 conventional_genes_sorted.txt sig_GF_sorted.txt > WT_only.txt
comm -13 conventional_genes_sorted.txt sig_GF_sorted.txt > GF_only.txt
```

Biological Insights: The shared genes between conventional and GF conditions suggest fundamental biological processes that are maintained regardless of the microbiota's presence. In contrast, the unique genes could be indicative of processes influenced by the microbiota.

- **Venn Diagram Creation:** Using a Python script (***vennefiles.py***) to visualize the overlap between GF and conventional genes, which is essential for understanding the extent of shared genetic responses between these two conditions.



- **Network Analysis:** With the shared.txt, WT_only.txt, and GF_only.txt files, you conducted a network analysis to explore the interactions between genes. By applying GO enrichment analysis to each group and selecting 1000 GO terms, and construct a network to visualize these interactions.

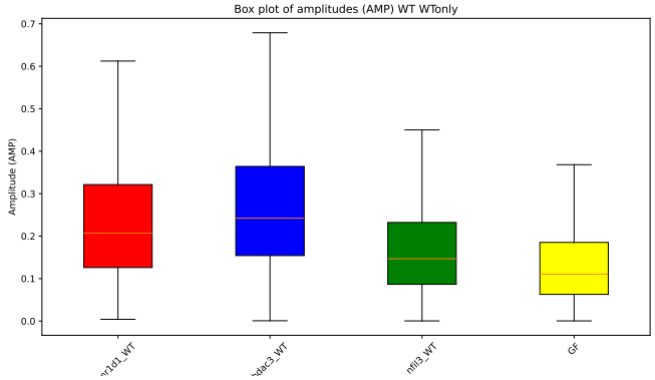


- **Circadian Feature Analysis**
  Comparison: The study compared WT, GF, and shared genes across the groups to understand circadian features.
  Tools: Python scripts were used to select relevant genes and analyze amplitude and phase differences. ***choose.py*** (input gene list .txt file and meta2d results file.) Select meta2d content based on the gene list file and use ***amp.py*** (give meta2d input files with the same number of genes)

AMP comparison: represent amplitude of rhythmic gene expression across different gene lists associated with wild type (WT) and germ-free (GF) conditions. Each box plot compares the amplitude of gene expression among the nr1d1, hdac3, nfil3 gene lists and GF conditions.

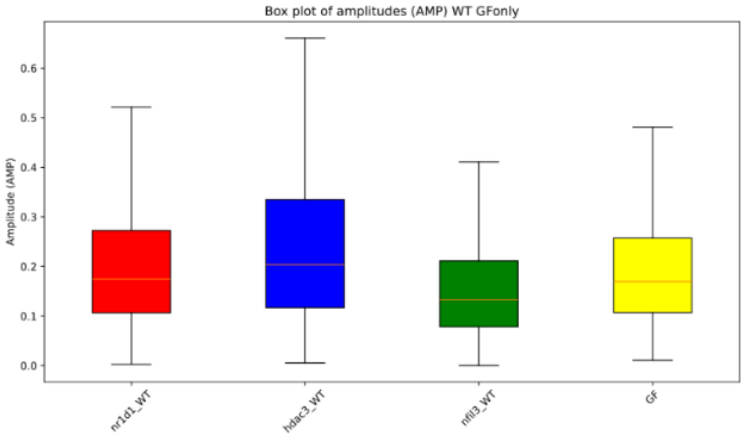
1) Box Plot of Amplitudes (AMP) WT Only

WT_only.txt meta2d_nr1d1_WT.txt meta2d_hdac3_WT.txt meta2d_nfil3_WT.txt meta2d_GF.txt



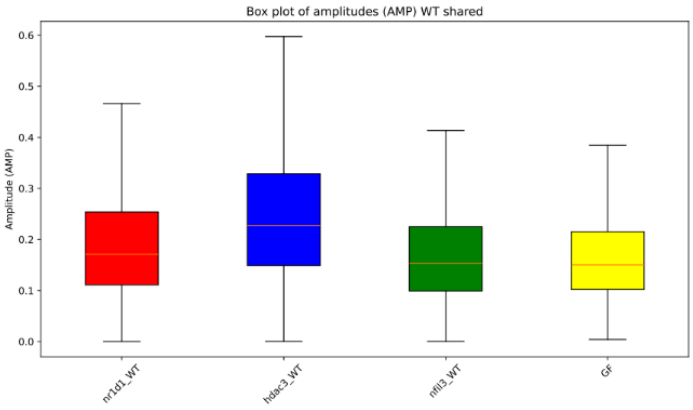
2) Box Plot of Amplitudes (AMP) GF Only

GF_only.txt meta2d_nr1d1_WT.txt meta2d_hdac3_WT.txt meta2d_nfil3_WT.txt meta2d_GF.txt



3) Box Plot of Amplitudes (AMP) shared

shared.txt meta2d_nr1d1_WT.txt meta2d_hdac3_WT.txt meta2d_nfil3_WT.txt meta2d_GF.txt

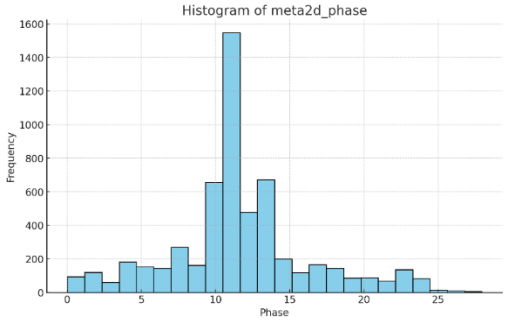


4) **Comparison Between Conditions**: The presence of the GF box in each plot allows for a direct comparison of gene expression amplitude between the GF and WT conditions. It looks like the amplitudes for GF are generally lower than for the WT conditions, which might suggest a dampening of rhythmic expression in the absence of a microbiota.

Phase comparison: Choose.py phase1.R
(I find something wrong with phase e.g. nr1d1,some phases greater than 24, may be the reason)

```
meta2d(infile="nr1d1_WT.txt",filestyle="txt", outdir="example_nr1d1_WT",
  , timepoints=seq(0, 42, by=6),
```

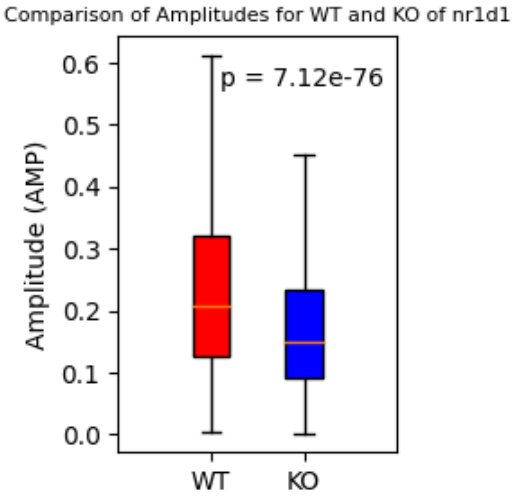
Couse there are replicated files for WT/KO, the total time 48h.



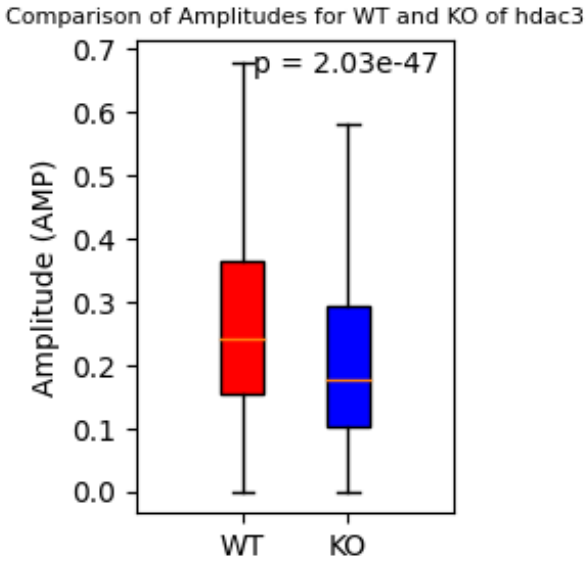
5. explore how the loss of specific genes

explore how the loss of specific genes—nr1d1, hdac3, and nfil3—affects circadian rhythms in conventional (CV) cycling genes. The approach involves a series of steps designed to identify gene dependencies, assess circadian features, and understand gene interactions within biological pathways.

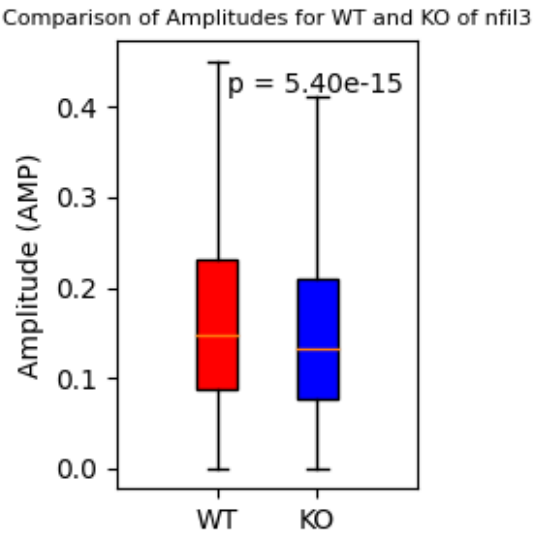
- **Impact on Circadian Rhythms:** analyze how the loss of each gene affects circadian rhythms by comparing gene expression amplitudes in wild type (WT) vs. knockout (KO) conditions for each gene using the ***choose.py*** and ***amp1.py*** scripts.
    1) **Nr1d1**
    wtonly_meta2d_nr1d1_WT.txt wtonly_meta2d_nr1d1_KO.txt



    2) **Hdac3**
    wtonly_meta2d_hdac3_WT.txt wtonly_meta2d_hdac3_KO.txt



    3) **Nfil3**
    wtonly_meta2d_nfil3_WT.txt wtonly_meta2d_nfil3_KO.txt



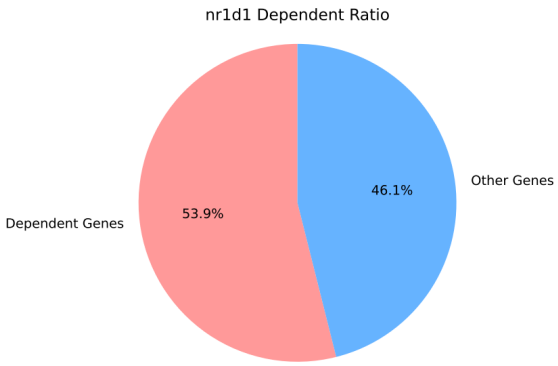
4) **Consistent Patterns:** All three genes show a consistent pattern where the knockout condition is associated with a decrease in the amplitude of circadian gene expression, suggesting that these genes play a role in maintaining the amplitude of circadian rhythms.

- **Circadian Features Analysis:**
  1) WT vs KO Comparison: The ***difference.***py(calculate the difference based on phase and amputitue between WT and KO),***modifydiffer.py***(if the phase difference value >12, we need use 24 – this value) and ***dependent.py***(find dependent genes) scripts are used to define Nr1d1, Hdac3, or Nfil3-dependent genes by identifying relative amplitude changes more than 2-fold or phase shifts greater than absolute 6-12 hours when comparing WT and KO.
  2) Gene List Generation: The awk command generates lists of dependent genes for each condition.

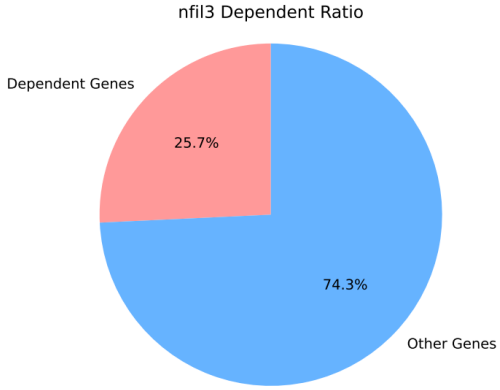
```Linux
awk '{print $1}' nfil3_dependent_modify.txt > nfil3_dependent_modify_genelist.txt
```

Group	counts
nr1d1_dependent_modify_genelist.txt	3061
hdac3_dependent_modify_genelist.txt	1607
nfil3_dependent_modify_genelist.txt	1462

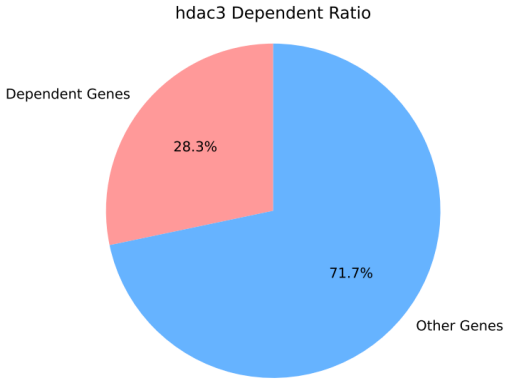
3) Pie Chart Visualization: The ***pie.py*** script creates pie charts to visualize the proportion of dependent vs. non-dependent genes.
   a. Nr1d1



b. Nfil3



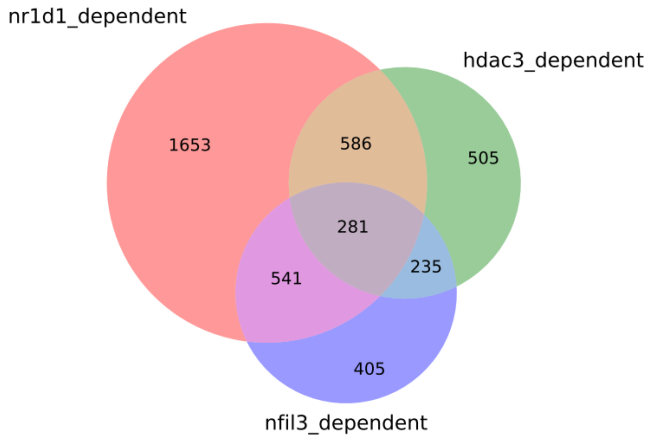
c. Hdac3



d. The differences in dependent gene proportions could be due to the varying functions of these genes in the circadian clock. nr1d1 appears to have a more central role, which could be related to its function as a transcriptional regulator in the clock mechanism.

- **Venn Diagram Creation:**

  Input files: nfil3_dependent_modify.txt hdac3_dependent_modify.txt nr1d1_dependent_modify.txt

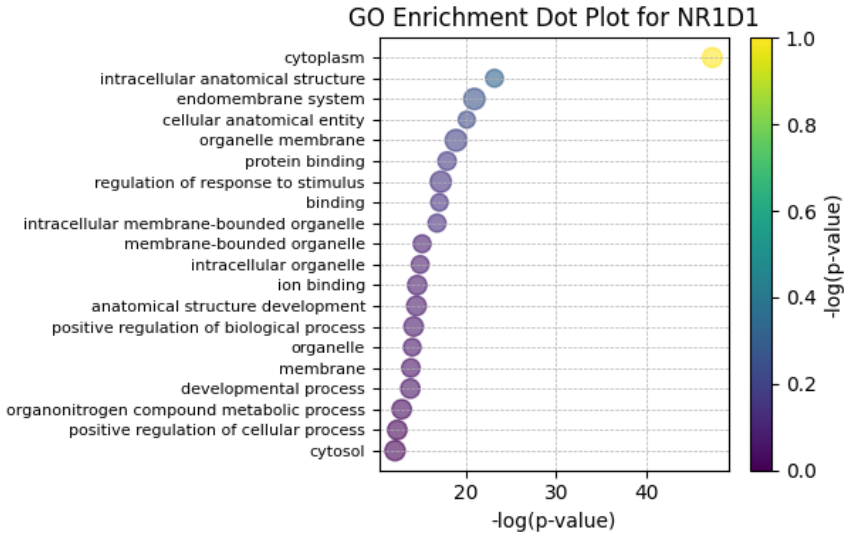


  the 281 genes at the center of the diagram suggest a core set of circadian-regulated genes that require the function of all three genes, indicating a potential combinatorial regulation or a central pathway requiring all three components.

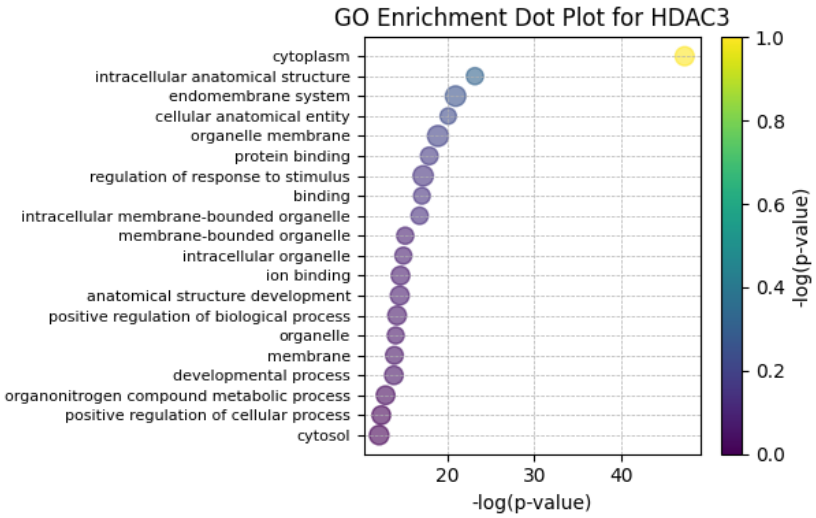
6.Pathway and Network Analysis:

- **GO Enrichment Analysis:** The ***GO.ipynb*** Jupyter notebook applies GO enrichment analysis to each set of dependent genes, creating dot plots that display the significance (color) and enrichment ratio (size) of GO terms.

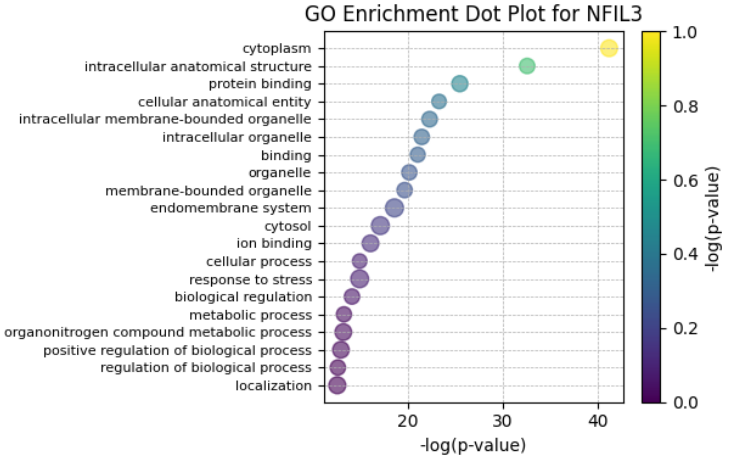
  1) Nr1d1



2) **Hdac3**

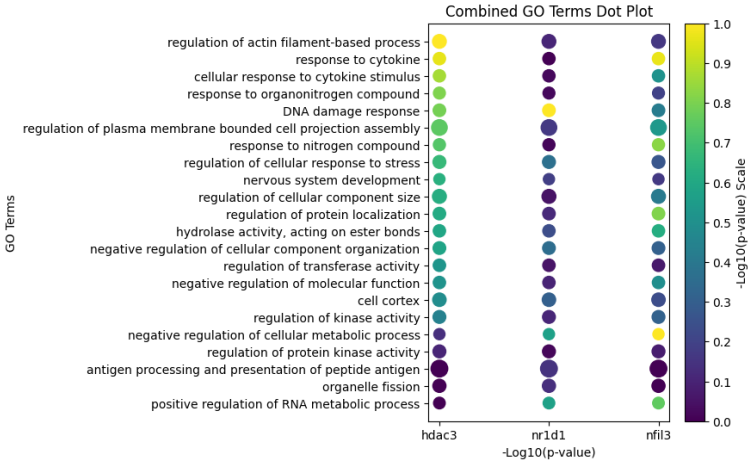


3) **Nfil3**

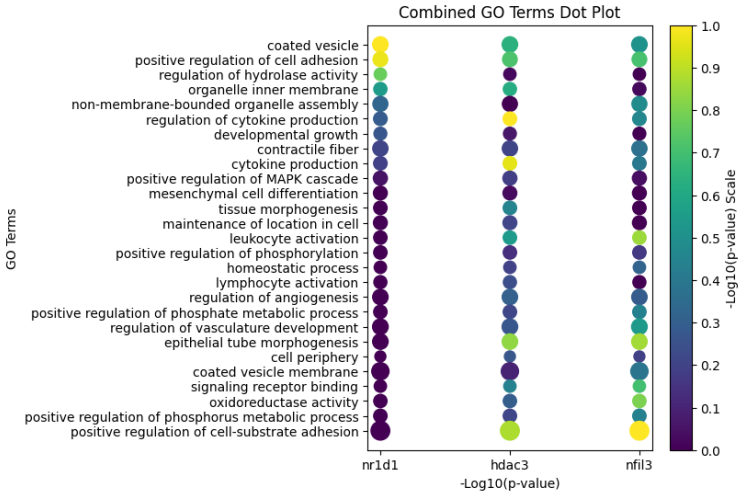


4) **Visualization and Interpretation**: The GO terms on the y-axis are sorted by the level of significance (p-value), represented on the x-axis as -log10(p-value). The color and size of the dots correspond to the significance and enrichment ratio, respectively, with more significant and more enriched terms appearing larger and more to the right on the plot. The terms closer to the top of each plot are the most significantly enriched in the respective gene set, indicating their potential biological importance

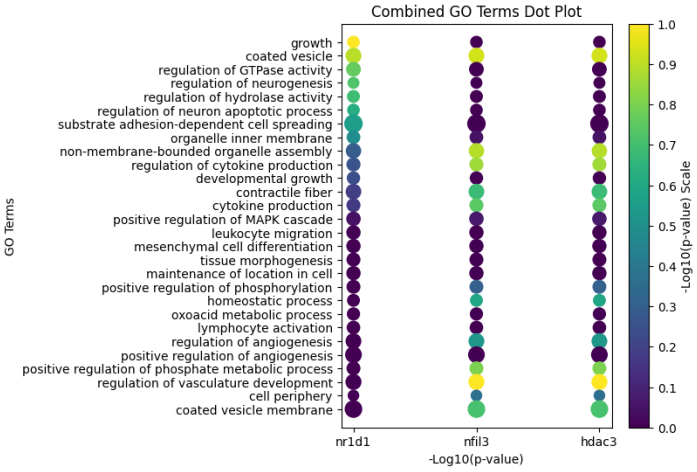
- **Combined GO Terms Dot Plot**
  1) **Unique nr1d1**



2) Unique hdac3

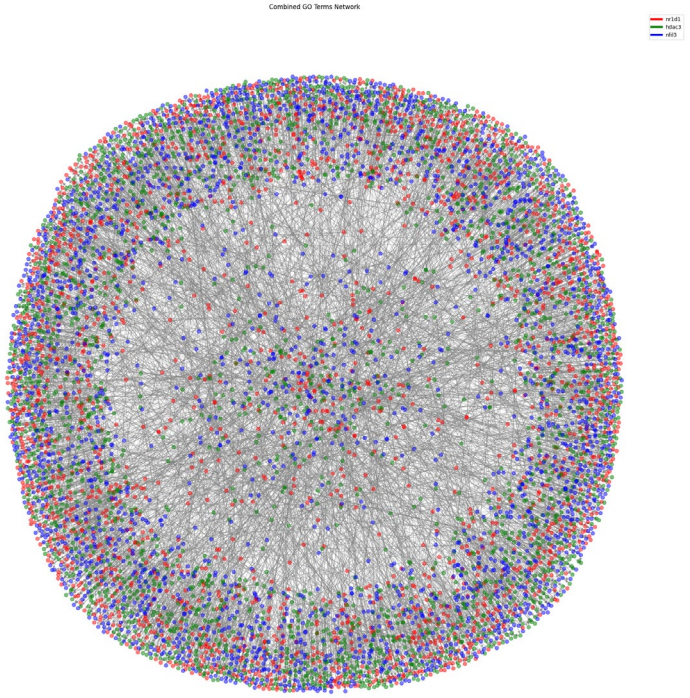


3) Unique nfil3



4) **Interrelated Functionality:** The combined plot integrates the GO terms from all three genes, illustrating the shared and unique biological functions associated with each gene set. It provides a comparative view of the functional enrichment across the gene sets, highlighting the unique and overlapping biological processes among them.

- **Network Visualization:** ***use network.ipynb.***The parents field from GO results is used to understand gene interactions, with a focus on 1000 GO terms for each gene set.Show relationships between GO terms based on the gene sets' enrichment data. It likely shows how these terms are interconnected, potentially revealing clusters of related functions or pathway.



7.Distance Analysis:

- **Distance Calculation**
  1) **Normalization:** python file ***normaliazediffer.py.*** Phase differences between gene pairs were normalized between 0 and 1 to standardize the data across all genes.

2) **Distance Measurement:** python file ***distance.py.*** For each gene pair, distances were calculated based on the normalized phase differences and amplitude differences, producing a quantitative measure of how similar or dissimilar the gene pairs are in terms of their circadian rhythm characteristics.

- **Similarity Analysis**
  1) **Similar Gene Pairs:** python file: ***similarity1.py.*** By setting a threshold distance (0.05), the analysis identifies gene pairs that are similar across the HDAC3, NR1D1, and NFIL3 datasets.
     e.g.:

```
Gene1,Gene2,Distance_hdac3,Distance_nr1d1,Distance_nfil3

ND5,Rps12-ps21,0.14915919102846828,0.17111609547848622,0.14424898897776045
```

  2) **Statistical Analysis:** python file: ***similarity2.py*** Mean distance, median, standard deviation, and variance are calculated for each dataset to describe the distribution of distances. An ANOVA test is then conducted to determine if the differences in distances between the datasets are statistically significant.
     a. Results:

```
Dataset,Mean,Median,Standard Deviation,Variance,ANOVA F-Statistic,ANOVA P-Value
HDAC3,0.3664401545769131 5,0.3201421966437215,0.2343358499022122,0.054913290549392126,343207.7
7991434536,0.0
NR1D1,0.3998537284465556 5,0.36323897381943127,0.2400452421195973,0.057621718264256085,343207.
77991434536,0.0
NFIL3,0.3305846945893071 7,0.27640098014195336,0.23016585747152737,0.052976321945603456,343207.
77991434536,0.0
```

     b. Analysis:The results showed highly significant differences between the datasets, as indicated by the very high F-statistics and a p-value of 0.0. This suggests that the gene pairs exhibit statistically different behaviors in terms of their circadian rhythm characteristics across the different conditions (HDAC3, NR1D1, and NFIL3). The significant differences in distances across the gene sets suggest that the circadian rhythms are distinctly regulated by these genes.
  3) **Network Construction:** python file: ***similarity3.py***
     a. **Graph Creation:** A network graph was created with nodes representing genes and edges representing the calculated distances between them. The weights of the edges correspond to the distances, indicating the degree of similarity.

The result is a GraphML file named "gene_network.graphml", which represents the combined network of gene-gene interactions from all three DataFrames. In this network, nodes represent genes, and edges represent the distance relationships between them. The weights of the edges in the GraphML file correspond to the 'distance' values from the data.

```
<?xml version='1.0' encoding='utf-8'?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaL
ocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key id="d0" for="edge" attr.name="weight" attr.type="double" />
  <graph edgedefault="undirected">
    <node id="ND5" />
    <node id="ND4L" />
    <node id="Pisd-ps3" />
    <node id="LOC100861738" />
    <node id="LOC100861749" />
    <node id="LOC434608" />
```

b. **Visualization Challenges:** Attempting to load this network into Cytoscape for visualization was problematic due to the large file size, which caused the software to shut down.