**APPENDIX A: DATA ANALYSIS AND VISUALIZATION CODE**

This appendix contains the Python scripts used to analyze the mutation screening data and generate visualizations for the report. The code includes data processing, aggregation, and plotting of the top five significant genes.

**Data Processing and Aggregation Code**

import os

os.chdir("../")

os.getcwd()

os.chdir("C:\\Users\\t05ae24\\OneDrive - University of Aberdeen/Computational Biotechnology Programming Assessment")

os.listdir()

import glob

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib as mpl

import random

import os

import numpy as np

import re

import seaborn as sbs

txt\_files = [f for f in os.listdir() if f.endswith(".txt")]

print(txt\_files)

**Reading one file path**

file\_path = txt\_files[0]

data = pd.read\_csv("Ah3\_data\_file.txt", sep="\t") #Reading one file to check and confirm the files structures

print(data.head())

print(data.columns)

print(data.dtypes)

print(data.isnull().sum())

**Getting a statistical summary of the one file with gene name Ah3**

**calculating the mean of the replicates for mRNA, Protein, and Cell Viability**

data["Mean\_mRNA\_WT"] = data[["mRNA.Expression.WT.Rep1", "mRNA.Expression.WT.Rep2", "mRNA.Expression.WT.Rep3"]].mean(axis=1)

data["Mean\_mRNA\_Mut"] = data[["mRNA.Expression.Mut.Rep1", "mRNA.Expression.Mut.Rep2", "mRNA.Expression.Mut.Rep3"]].mean(axis=1)

data["Mean\_Protein\_WT"] = data[["Protein.Expression.WT.Rep1", "Protein.Expression.WT.Rep2", "Protein.Expression.WT.Rep3"]].mean(axis=1)

data["Mean\_Protein\_Mut"] = data[["Protein.Expression.Mut.Rep1", "Protein.Expression.Mut.Rep2", "Protein.Expression.Mut.Rep3"]].mean(axis=1)

data["Mean\_CellViability\_WT"] = data[["CellViability.WT.Rep1", "CellViability.WT.Rep2", "CellViability.WT.Rep3"]].mean(axis=1)

data["Mean\_CellViability\_Mut"] = data[["CellViability.Mut.Rep1", "CellViability.Mut.Rep2", "CellViability.Mut.Rep3"]].mean(axis=1)

print(data[["Gene", "Mean\_mRNA\_WT", "Mean\_mRNA\_Mut", "Mean\_Protein\_WT", "Mean\_Protein\_Mut", "Mean\_CellViability\_WT", "Mean\_CellViability\_Mut"]].head())

**Reading all Datasets**

all\_data = []

for file\_path in txt\_files:

data = pd.read\_csv(file\_path, sep="\t")

all\_data.append(data)

combined\_data = pd.concat(all\_data, ignore\_index=True)

print(combined\_data.head())

print(combined\_data.shape)

pd.set\_option("display.max\_rows", None)

pd.set\_option("display.max\_rows", None)

combined\_data.reset\_index(drop=True, inplace=True)

combined\_data.index += 1

print(combined\_data)

**Calculation of Mean of the replicates for both Wildtype and Mutant mRNA, Protein, and Cell Viability**

combined\_data["Mean\_mRNA\_WT"] = combined\_data[["mRNA.Expression.WT.Rep1", "mRNA.Expression.WT.Rep2", "mRNA.Expression.WT.Rep3"]].mean(axis=1)

combined\_data["Mean\_mRNA\_Mut"] = combined\_data[["mRNA.Expression.Mut.Rep1", "mRNA.Expression.Mut.Rep2", "mRNA.Expression.Mut.Rep3"]].mean(axis=1)

combined\_data["Mean\_Protein\_WT"] = combined\_data[["Protein.Expression.WT.Rep1", "Protein.Expression.WT.Rep2", "Protein.Expression.WT.Rep3"]].mean(axis=1)

combined\_data["Mean\_Protein\_Mut"] = combined\_data[["Protein.Expression.Mut.Rep1", "Protein.Expression.Mut.Rep2", "Protein.Expression.Mut.Rep3"]].mean(axis=1)

combined\_data["Mean\_CellViability\_WT"] = combined\_data[["CellViability.WT.Rep1", "CellViability.WT.Rep2", "CellViability.WT.Rep3"]].mean(axis=1)

combined\_data["Mean\_CellViability\_Mut"] = combined\_data[["CellViability.Mut.Rep1", "CellViability.Mut.Rep2", "CellViability.Mut.Rep3"]].mean(axis=1)

print(combined\_data[["Gene", "Mean\_mRNA\_WT", "Mean\_mRNA\_Mut", "Mean\_Protein\_WT", "Mean\_Protein\_Mut", "Mean\_CellViability\_WT", "Mean\_CellViability\_Mut"]].head())

**Renaming Columns**:

combined\_data = combined\_data.rename(columns={ "Gene": "GENE NAME", "WildType.Sequence": "WILDTYPE GENE DNA SEQUENCE", "Mutant.Sequence": "MUTANT GENE DNA SEQUENCE","Mean\_mRNA\_WT": "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "Mean\_mRNA\_Mut": "MEAN NORMALISED mRNA EXPRESSION MUTANT", "Mean\_Protein\_WT": "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "Mean\_Protein\_Mut": "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "Mean\_CellViability\_WT": "MEAN NORMALISED CELL VIABILITY WILDTYPE", "Mean\_CellViability\_Mut": "MEAN NORMALISED CELL VIABILITY MUTANT"})

print(combined\_data.columns)

print(combined\_data[[ "GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT" ]].head())

result\_table = combined\_data[["GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT" ]]

combined\_data.reset\_index(drop=True, inplace=True)

combined\_data.index += 1

display(result\_table)

result\_table.to\_csv("Mutation\_Analysis\_Results.txt", sep="\t", index=False)

**Mutation Type and Location Identification:**

def get\_mutation\_type(wildtype\_seq, mutant\_seq):

if len(wildtype\_seq) == len(mutant\_seq):

if wildtype\_seq != mutant\_seq:

return "Substitution"

elif len(wildtype\_seq) < len(mutant\_seq):

return "Insertion"

elif len(wildtype\_seq) > len(mutant\_seq):

return "Deletion"

return "No Mutation"

combined\_data["MUTATION TYPE"] = combined\_data.apply (lambda row: get\_mutation\_type(row["WILDTYPE GENE DNA SEQUENCE"], row["MUTANT GENE DNA SEQUENCE"]), axis=1)

combined\_data["MUTATION LOCATION"] = combined\_data.apply (lambda row: "Promoter" if re.match(r"^.{0,1000}$", row["MUTANT GENE DNA SEQUENCE"]) else "Coding", axis=1)

print(combined\_data[["GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT", "MUTATION TYPE", "MUTATION LOCATION"]])

combined\_data[["GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT", "MUTATION TYPE", "MUTATION LOCATION"]].to\_csv("Mutation\_Type\_and\_Location\_Results.txt", sep="\t", index=False)

**Expression Level Calculations:**

summary\_df = combined\_data [["GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT", "MUTATION TYPE", "MUTATION LOCATION"]].copy()

summary\_df["MEAN NORMALISED mRNA EXPRESSION WILDTYPE"] = combined\_data["MEAN NORMALISED mRNA EXPRESSION WILDTYPE"]

summary\_df["MEAN NORMALISED mRNA EXPRESSION MUTANT"] = combined\_data["MEAN NORMALISED mRNA EXPRESSION MUTANT"]

summary\_df["MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE"] = combined\_data["MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE"]

summary\_df["MEAN NORMALISED PROTEIN EXPRESSION MUTANT"] = combined\_data["MEAN NORMALISED PROTEIN EXPRESSION MUTANT"]

summary\_df["MEAN NORMALISED CELL VIABILITY WILDTYPE"] = combined\_data["MEAN NORMALISED CELL VIABILITY WILDTYPE"]

summary\_df["MEAN NORMALISED CELL VIABILITY MUTANT"] = combined\_data["MEAN NORMALISED CELL VIABILITY MUTANT"]

summary\_df["FOLDCHANGE mRNA"] = summary\_df["MEAN NORMALISED mRNA EXPRESSION MUTANT"] / summary\_df["MEAN NORMALISED mRNA EXPRESSION WILDTYPE"]

summary\_df["FOLDCHANGE PROTEIN"] = summary\_df["MEAN NORMALISED PROTEIN EXPRESSION MUTANT"] / summary\_df["MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE"]

summary\_df["FOLDCHANGE CELLVIABILITY"] = summary\_df["MEAN NORMALISED CELL VIABILITY MUTANT"] / summary\_df["MEAN NORMALISED CELL VIABILITY WILDTYPE"]

print(summary\_df.head())

display(summary\_df)

summary\_df.to\_csv("All\_Gene\_Summary.txt", sep="\t", index=False)

print(summary\_df.columns.tolist())

**Effect of Fold Change**:

def categorize\_change(fold\_change):

if fold\_change > 1.01: # Slight buffer to avoid noise

return "Increase"

elif fold\_change < 0.99:

return "Decrease"

else: return "No Effect"

summary\_df["EFFECT ON mRNA EXPRESSION LEVEL"] = summary\_df["FOLDCHANGE mRNA"].apply(categorize\_change)

summary\_df["EFFECT ON PROTEIN EXPRESSON LEVEL"] = summary\_df["FOLDCHANGE PROTEIN"].apply(categorize\_change)

summary\_df["EFFECT ON CELLVIABILITY"] = summary\_df["FOLDCHANGE CELLVIABILITY"].apply(categorize\_change)

print(summary\_df[["GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT", "MUTATION TYPE", "MUTATION LOCATION", "FOLDCHANGE mRNA", "EFFECT ON mRNA EXPRESSION LEVEL", "FOLDCHANGE PROTEIN", "EFFECT ON PROTEIN EXPRESSON LEVEL", "FOLDCHANGE CELLVIABILITY", "EFFECT ON CELLVIABILITY"]])

display(summary\_df)

columns = summary\_df.columns.tolist()

new\_columns = [ "GENE NAME", "WILDTYPE GENE DNA SEQUENCE", "MUTANT GENE DNA SEQUENCE", "MEAN NORMALISED mRNA EXPRESSION WILDTYPE", "MEAN NORMALISED mRNA EXPRESSION MUTANT", "MEAN NORMALISED PROTEIN EXPRESSION WILDTYPE", "MEAN NORMALISED PROTEIN EXPRESSION MUTANT", "MEAN NORMALISED CELL VIABILITY WILDTYPE", "MEAN NORMALISED CELL VIABILITY MUTANT", "MUTATION TYPE", "MUTATION LOCATION", "FOLDCHANGE mRNA", "EFFECT ON mRNA EXPRESSION LEVEL", "FOLDCHANGE PROTEIN", "EFFECT ON PROTEIN EXPRESSON LEVEL", "FOLDCHANGE CELLVIABILITY", "EFFECT ON CELLVIABILITY" ]

summary\_df = summary\_df[new\_columns]

display(summary\_df)

summary\_df.to\_csv("All\_Gene\_Summary.txt", sep="\t", index=False)

**Filtering Significant Mutations:**

summary\_df["FOLD CHANGE COMBINED IMPACT SCORE"] = (summary\_df["FOLDCHANGE CELLVIABILITY"] - 1).abs() + \

(summary\_df["FOLDCHANGE mRNA"] - 1).abs() + \

(summary\_df["FOLDCHANGE PROTEIN"] - 1).abs()

TOP\_FIVE\_GENES = summary\_df.sort\_values(by="FOLD CHANGE COMBINED IMPACT SCORE", ascending=False).head(5)

print(TOP\_FIVE\_GENES)

display(TOP\_FIVE\_GENES)

TOP\_FIVE\_GENES.to\_csv("Top\_5\_Significant\_Genes.txt", sep="\t", index=False)

**Statistical Analysis and Data Visualization**

**Box Plot:**

sbs.set(style="whitegrid")

plt.figure(figsize=(8, 6))

sbs.boxplot(x="MUTATION TYPE", y="MEAN NORMALISED mRNA EXPRESSION WILDTYPE", data=TOP\_FIVE\_GENES)

plt.title("Boxplot of MEAN NORMALISED mRNA EXPRESSION WILDTYPE by Mutation Type")

plt.xticks(rotation=45)

plt.xlabel("MUTATION TYPE")

plt.ylabel("MEAN NORMALISED mRNA EXPRESSION WILDTYPE")

plt.yscale("log")

handles = [ plt.Line2D([0], [0], color="tab:orange", lw=4, label="Deletion"), plt.Line2D([0], [0], color="tab:blue", lw=4, label="Substitution"), plt.Line2D([0], [0], color="tab:green", lw=4, label="Insertion") ]

plt.legend(handles=handles, loc="upper right", title="MUTATION TYPES")

plt.tight\_layout()

plt.savefig("Boxplot of MEAN NORMALISED mRNA EXPRESSION WILDTYPE by Mutation Type")

plt.show()

**Scatter Plots:**

mutation\_colors = { "Deletion": "orange", "Substitution": "blue", "Insertion": "green" }

colors = [mutation\_colors[mut] for mut in TOP\_FIVE\_GENES["MUTATION TYPE"]]

fig, ax = plt.subplots(figsize=(5, 5))

scatter = ax.scatter(TOP\_FIVE\_GENES["MEAN NORMALISED mRNA EXPRESSION MUTANT"], TOP\_FIVE\_GENES["MEAN NORMALISED PROTEIN EXPRESSION MUTANT"], c=colors, alpha=0.7, edgecolors="black")

ax.set\_xlabel("MEAN Normalised mRNA Expression (Mutant)")

ax.set\_ylabel("MEAN Normalised Protein Expression (Mutant)")

ax.set\_title("Scatter Plot: MEAN Normalised mRNA Expression(Mutant)\nvs MEAN Normalised Protein Expression (Mutant)")

ax.set\_xscale("log")

ax.set\_yscale("log")

from matplotlib.lines import Line2D

legend\_elements = [Line2D([0], [0], marker="o", color="w", markerfacecolor=color, markersize=8, label=mut) for mut, color in mutation\_colors.items()]

ax.legend(handles=legend\_elements, title="Mutation Type", loc="upper left")

ax.grid(True, linestyle="--", alpha=0.4)

plt.tight\_layout()

plt.savefig("Scatter\_plot of MEAN Normalised mRNA Expression (Mutant) MEAN Normalised Protein Expression (Mutant)")

plt.show()

**Histogram Density Plots**:

fig, ax = plt.subplots(figsize=(8, 5))

ax.hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Deletion"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Deletion", color="orange", edgecolor="black")

ax.hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Substitution"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Substitution", color="blue", edgecolor="black")

ax.hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Insertion"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Insertion", color="green", edgecolor="black")

ax.set\_xlabel("MEAN Normalised Cell Viability (Mutant)")

ax.set\_ylabel("Frequency")

ax.set\_title("Histogram: MEAN Normalised Cell Viability by Mutation Type")

ax.legend(title="Mutation Type")

plt.savefig("Histogram Frequency Plot of MEAN Normalised Cell Viability")

plt.show()

**Multi-panel plots**

fig, ax = plt.subplot\_mosaic([["box", "scatter"], ["density", "density"]],

figsize=(8, 8), layout="constrained")

sbs.boxplot(data=TOP\_FIVE\_GENES, x="MUTATION TYPE", y="MEAN NORMALISED mRNA EXPRESSION WILDTYPE", ax=ax["box"])

ax["box"].set\_title("MEAN NORMALISED mRNA EXPRESSION WILDTYPE\nvs Mutation Type")

ax["box"].set\_xlabel("Mutation Type")

ax["box"].set\_ylabel("MEAN Normalised mRNA Expression (Wildtype)")

ax["box"].set\_yscale("log")

ax["box"].legend(handles, ["Deletion", "Substitution", "Insertion"], title="Mutation Type", loc="upper right", fontsize=8, title\_fontsize=9)

sbs.scatterplot(data=TOP\_FIVE\_GENES, x="MEAN NORMALISED PROTEIN EXPRESSION MUTANT",

y="MEAN NORMALISED mRNA EXPRESSION MUTANT", hue="MUTATION TYPE", ax=ax["scatter"], palette={"Deletion": "orange", "Substitution": "blue", "Insertion": "green"})

ax["scatter"].set\_title("MEAN Normalised mRNA Expression (Mutant)\nvs MEAN Normalised Protein Expression (Mutant)")

ax["scatter"].set\_xlabel("MEAN Normalised mRNA Expression (Mutant)")

ax["scatter"].set\_ylabel("MEAN Normalised Protein Expression (Mutant)")

ax["scatter"].legend(title="Mutation Type", fontsize=8, title\_fontsize=9)

ax["scatter"].set\_xscale("log")

ax["scatter"].set\_yscale("log")

ax["density"].hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Deletion"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Deletion", color="orange", edgecolor="black")

ax["density"].hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Substitution"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Substitution", color="blue", edgecolor="black")

ax["density"].hist(TOP\_FIVE\_GENES[TOP\_FIVE\_GENES["MUTATION TYPE"] == "Insertion"]["MEAN NORMALISED CELL VIABILITY MUTANT"], bins=5, alpha=0.5, label="Insertion", color="green", edgecolor="black")

ax["density"].set\_xlabel("MEAN Normalised Cell Viability (Mutant)")

ax["density"].set\_ylabel("Frequency")

ax["density"].set\_title("Histogram: MEAN Normalised Cell Viability by Mutation Type")

ax["density"].legend(title="Mutation Type", fontsize=8, title\_fontsize=9)

plt.savefig("Top five Genes Plot")

plt.show()