

```

import numpy as np
import pandas as pd

import statsmodels.api as sm
import matplotlib.pyplot as plt

import seaborn as sns

# Step 1: Simulate RNA-Seq Dataset
np.random.seed(42)

genes = [f'gene_{i}' for i in range(1, 101)]
conditions = ['Control', 'Treatment']

samples = [f'sample_{i}' for i in range(1, 11)]
data = np.random.poisson(lam=20, size=(100, 10))

# Simulate differential expression for some genes in Treatment condition
data[0:5, 5:10] += 15

# Create DataFrame
df = pd.DataFrame(data, index=genes, columns=samples)

metadata = pd.DataFrame({'sample': samples,
                        'condition': ['Control']*5 + ['Treatment']*5})

# Step 2: Normalize the Data
df_norm = df.div(df.sum(axis=0), axis=1) * 10**6
df_log = np.log2(df_norm + 1)

def differential_expression(df, metadata):
    results = []
    for gene in df.index:
        y = df_log.loc[gene].values
        X = pd.get_dummies(metadata['condition'], drop_first=True)
        # The line below was modified to cast the DataFrame to float
        X = sm.add_constant(X.astype(float))
        model = sm.OLS(y, X).fit()
        p_value = model.pvalues[1]
        results.append({'gene': gene, 'p_value': p_value})

    results_df = pd.DataFrame(results)
    results_df['adjusted_p_value'] = sm.stats.multipletests(results_df['p_value'], method='fdr_bh')[1]

    return results_df

# Call the differential_expression function to calculate results_df
results_df = differential_expression(df_log, metadata) # This line was added to call the function

# Filter differentially expressed genes
deg = results_df[results_df['adjusted_p_value'] < 0.05]

# Step 4: Functional Annotation (Simulated Annotations)
annotations = {
    'gene_1': 'Pathway A',
    'gene_2': 'Pathway B',
    'gene_3': 'Pathway C',
    'gene_4': 'Pathway D',
    'gene_5': 'Pathway E',
}
deg['annotation'] = deg['gene'].map(annotations).fillna('Unknown')

# Step 5: Biological Interpretation (Plotting)
plt.figure(figsize=(10, 6))

sns.scatterplot(x='gene', y='adjusted_p_value', hue='annotation', data=deg)
plt.axhline(y=0.05, color='r', linestyle='--')

plt.xlabel('Genes')
plt.ylabel('Adjusted P-Value')

plt.title('Differentially Expressed Genes')
plt.xticks(rotation=90)

plt.legend(title='Annotations')
plt.tight_layout()

plt.show()

# Save results to a CSV file
deg.to_csv('differentially_expressed_genes.csv', index=False)

```

```
# Generate the Report
report = f"""
RNA-Seq Data Analysis Report

Differentially Expressed Genes

{deg[['gene', 'adjusted_p_value']]}

Functional Annotations

{deg[['gene', 'annotation']]}

Potential Biological Interpretations

The genes gene_1, gene_2, etc., are involved in pathways A, B, etc.
These pathways are important for understanding the effect of the treatment condition.
"""

# Save the report to a text file
with open('RNASeq_Analysis_Report.txt', 'w') as f:
    f.write(report)

print("Analysis complete. Results saved to 'differentially_expressed_genes.csv' and 'RNASeq_Analysis_Report.txt'.")

Start coding or generate with AI.
```