**Bioinformatics Programming: Comprehensive Code Review**

**Module 1: R Basics**

**Working with R and RStudio**

r

Copy

*# Check working directory*

getwd()

setwd("path/to/your/directory")

*# Basic operations*

a <- 1 *# Assignment*

b <- 2

print(a + b)

*# Vectors*

v1 <- c(1, 2, 3, 6, 7, 8)

v2 <- c(3:8) *# Sequence*

length(v1) *# Get length*

v1 + v2 *# Vector operations*

*# Lists*

lst <- list(1, 2, "three")

nmd.lst <- list(first = 1, second = 2, third = "three")

nmd.lst$second *# Access by name*

*# Matrices*

M1 <- matrix(0, 2, 3) *# Create 2x3 matrix*

dim(M1) *# Get dimensions*

M1[1,1] <- 1 *# Assignment*

*# Data Frames*

df1 <- data.frame(

col1 = c(1,2,3),

col2 = c(2,3,4),

name = c("Gordon","Maria","Daniel")

)

**File Operations**

r

Copy

*# Read CSV*

data <- read.csv("data.csv", row.names = 1)

*# Write CSV*

write.csv(data, "output.csv")

*# Read TSV*

tab\_data <- read.table("data.txt", sep="\t")

*# Save R objects*

save(data, file="analysis.RData")

load("analysis.RData")

**Module 2: Advanced R Programming**

**Logical Operations**

r

Copy

*# Basic logical operations*

x > 3 *# Greater than*

x >= 3 *# Greater than or equal*

x < 3 *# Less than*

x <= 3 *# Less than or equal*

x == 3 *# Equal to*

x != 3 *# Not equal to*

*# Combining conditions*

condition1 & condition2 *# AND*

condition1 | condition2 *# OR*

!condition1 *# NOT*

**Statistical Analysis**

r

Copy

*# Correlation analysis*

correlation <- cor(x, y)

cor\_test <- cor.test(x, y)

cor\_test\_positive <- cor.test(x, y, alternative="greater")

*# Linear Models*

model <- lm(y ~ x, data=df)

summary(model)

residuals(model)

fitted(model)

*# T-tests*

t.test(group1, group2)

*# ANOVA*

result <- aov(value ~ group, data=df)

summary(result)

**Data Visualization**

r

Copy

library(ggplot2)

*# Basic scatter plot*

ggplot(data, aes(x=x, y=y)) +

geom\_point()

*# Box plot*

ggplot(data, aes(x=group, y=value)) +

geom\_boxplot()

*# Histogram*

ggplot(data, aes(x=value)) +

geom\_histogram()

**Module 3: Python for Bioinformatics**

python

Copy

import pandas as pd

import numpy as np

import scanpy as sc

import anndata as ad

*# Read data*

adata = sc.read\_h5ad("data.h5ad")

*# Basic preprocessing*

sc.pp.filter\_cells(adata, min\_genes=200)

sc.pp.filter\_genes(adata, min\_cells=3)

*# Normalize data*

sc.pp.normalize\_total(adata, target\_sum=1e4)

sc.pp.log1p(adata)

*# Find variable genes*

sc.pp.highly\_variable\_genes(adata)

**Module 4: Bioinformatics Applications**

**Bulk RNA-seq Analysis (R)**

r

Copy

library(DESeq2)

*# Create DESeq dataset*

dds <- DESeqDataSetFromMatrix(

countData = counts,

colData = coldata,

design = ~ condition

)

*# Run analysis*

dds <- DESeq(dds)

res <- results(dds)

*# Filter results*

sig\_genes <- res[!is.na(res$padj) & res$padj < 0.05, ]

**Single-cell RNA-seq Analysis (Python)**

python

Copy

*# Quality control*

adata = sc.read\_h5ad("data.h5ad")

sc.pp.calculate\_qc\_metrics(adata)

*# Preprocessing*

sc.pp.normalize\_total(adata)

sc.pp.log1p(adata)

sc.pp.highly\_variable\_genes(adata)

*# Dimensionality reduction*

sc.tl.pca(adata)

sc.pp.neighbors(adata)

sc.tl.umap(adata)

*# Clustering*

sc.tl.leiden(adata)

*# Find markers*

sc.tl.rank\_genes\_groups(adata, 'leiden')

**Integration Example**

r

Copy

*# R: Export bulk results*

write.csv(deseq\_results, "bulk\_results.csv")

*# Python: Import and integrate*

import pandas as pd

bulk\_results = pd.read\_csv("bulk\_results.csv")

sc\_results = pd.read\_csv("sc\_results.csv")

integrated = pd.merge(

bulk\_results,

sc\_results,

on="gene\_id"

)

This review covers the main coding aspects from all modules, with practical examples and common use cases in bioinformatics analysis.