

```

# =====
# FINAL VERIFIED SCRIPT: Single-Cell RNA-Seq Analysis (PBMC 3k)
# =====

# 1. ENVIRONMENT SETUP
# -----
work_dir <- "D:/DOWNLOADS"

# Create and set directory
if (!dir.exists(work_dir)) {
  dir.create(work_dir, recursive = TRUE)
}
setwd(work_dir)
print(paste("Running in:", getwd()))

# Load required libraries
library(Seurat)
library(dplyr)
library(patchwork)
library(ggplot2)
library(plotly)
library(htmlwidgets)

# 2. DATA DOWNLOAD & IMPORT
# -----
print("--- Step 2: Loading Data ---")
data_sub_dir <- file.path(work_dir, "pbmc3k_data")
tar_file <- file.path(work_dir, "pbmc3k.tar.gz")

# Download if missing
if (!file.exists(tar_file)) {
  print("Downloading PBMC 3k dataset...")

download.file("https://cf.10xgenomics.com/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz", destfile = tar_file)
  untar(tar_file, exdir = data_sub_dir)
}

# Read 10x Data
pbmc.data <- Read10X(data.dir = file.path(data_sub_dir,
"filtered_gene_bc_matrices/hg19"))
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)

# 3. QC & FILTERING
# -----
print("--- Step 3: Quality Control ---")
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")

# Save QC Violin Plot
vln_plot <- VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA",
"percent.mt"), ncol = 3)
ggsave("01_QC_Violin.png", vln_plot, width = 10, height = 6)

# Filter low quality cells
pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 &
percent.mt < 5)

# 4. NORMALIZATION & CLUSTERING
# -----
print("--- Step 4: Analysis Pipeline ---")
pbmc <- NormalizeData(pbmc)
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
pbmc <- ScaleData(pbmc)

```

```

pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))

# Save Elbow Plot
ggsave("02_ElbowPlot.png", ElbowPlot(pbmc), width = 6, height = 4)

# Cluster Cells
pbmc <- FindNeighbors(pbmc, dims = 1:10)
pbmc <- FindClusters(pbmc, resolution = 0.5)

# Run standard 2D UMAP
pbmc <- RunUMAP(pbmc, dims = 1:10)
umap_static <- DimPlot(pbmc, reduction = "umap", label = TRUE) + ggtitle("PBMC
3k Clusters (Unlabeled)")
ggsave("03_UMAP_Clusters_Static.png", umap_static, width = 8, height = 6)

# 5. INTERACTIVE 2D VISUALIZATION
# -----
print("--- Step 5: Generating Interactive 2D Plots ---")
# Interactive Clusters
interactive_umap <- ggplotly(umap_static)
saveWidget(interactive_umap, file = file.path(work_dir,
"04_Interactive_UMAP.html"))

# Interactive Gene Expression (Example: B-Cells)
gene_plot <- FeaturePlot(pbmc, features = "MS4A1") + ggtitle("MS4A1 Expression
(B-Cells)")
interactive_gene <- ggplotly(gene_plot)
saveWidget(interactive_gene, file = file.path(work_dir,
"05_Interactive_Gene_MS4A1.html"))

# 6. INTERACTIVE 3D VISUALIZATION
# -----
print("--- Step 6: Generating Interactive 3D Cube ---")
# Calculate specific 3D UMAP coordinates
pbmc <- RunUMAP(pbmc, dims = 1:10, n.components = 3L, reduction.name = "umap3d")
umap_3d_data <- FetchData(pbmc, vars = c("umap3d_1", "umap3d_2", "umap3d_3",
"seurat_clusters"))

# Create 3D Plot
plot_3d <- plot_ly(data = umap_3d_data,
                     x = ~umap3d_1, y = ~umap3d_2, z = ~umap3d_3,
                     color = ~seurat_clusters, colors = "Set1",
                     type = "scatter3d", mode = "markers",
                     marker = list(size = 3, opacity = 0.8)) %>%
  layout(title = "3D Interactive UMAP Clusters")

saveWidget(plot_3d, file = file.path(work_dir, "06_Interactive_3D_UMAP.html"))

# 7. MARKER IDENTIFICATION
# -----
print("--- Step 7: Saving Marker Data ---")
pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25,
logfc.threshold = 0.25)
write.csv(pbmc.markers, file = file.path(work_dir, "07_Cluster_Markers.csv"))

# Save Heatmap
top10 <- pbmc.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)
heatmap <- DoHeatmap(pbmc, features = top10$gene) + NoLegend()
ggsave("08_Heatmap_Top10.png", heatmap, width = 14, height = 10)

# 8. FINAL ANNOTATION (Renaming Clusters)
# -----
print("--- Step 8: Applying Biological Labels ---")
new.cluster.ids <- c("Naive CD4 T", "Memory CD4 T", "CD14+ Mono", "B", "CD8 T",

```

```
"FCGR3A+", "NK", "DC", "Platelet")
names(new.cluster.ids) <- levels(pbmc)
pbmc <- RenameIdents(pbmc, new.cluster.ids)

# Save Final Annotated Plot
final_plot <- DimPlot(pbmc, reduction = "umap", label = TRUE, pt.size = 0.5) +
  ggtile("PBMC 3k: Final Annotated Cell Types") +
  theme(plot.title = element_text(hjust = 0.5))
ggsave("09_Final_Annotated_UMAP.png", final_plot, width = 10, height = 7)

# 9. SAVE PROJECT
# -----
saveRDS(pbmc, file = file.path(work_dir, "pbmc3k_annotated_final.rds"))

print("====")
print("VERIFIED SUCCESS! All files saved to D:/DOWNLOADS")
print("=====")
```