Supplemental File 2: API calls for Seurat-based functions.
Briefly, interactive visualizations require the following classes from the input Seurat object: reductions, meta.data, version, assays, active.assay, and active.ident. The `export_shiny_object()` function within the cellcuratoR package extracts the aforementioned classes from the input Seurat object and saves them into a smaller S4 object to be interpreted by the interactive interface.

Dimensionality reduction plot
The dimensionality reduction visualization is generated with the Seurat function `DimPlot()`. By default, the `export_shiny_object()` function extracts the UMAP dimensionality reduction attribute from the data slot seurat_object@reductions. If `RunUMAP()` dimensionality reduction has not been run on the input seurat_object, then the tSNE dimensionality reduction method is extracted before the PCA dimensionality reduction method. The resulting plot is made interactive with the R package plotly (v4.9.0), allowing the user to zoom in on subpopulations of cells and hover over cells to determine their cluster identity. Cells in the plot can be colored according to cluster identity (`Idents(seurat_obj) <- "final_cluster_label"`) or by originating library (`Idents(seurat_obj) <- "libraryID"`).

Heatmaps
Gene expression can be visualized across clustered cells in the form of heatmaps, which are generated with the Seurat function `FeaturePlot()`. By default, cells with higher expression are colored in darker shades of blue. The custom scale within cellcuratoR depicts expression level in terms of transcripts per 10,000 (TP10K), as the input data is log normalized with a scale-factor of 10,000.

Violin Plots
Violin plots depicting the expression of a gene of interest in each cluster are constructed with the cellcuratoR functions `prepare_violin_data_colors()` and `construct_violin_plot()`. Expression distributions are only drawn if at least 25% of cells in a cluster express the gene of interest.

Differential Expression
First, the identity of the seurat_object is set according to the type of comparison indicated by the user (eg, `Idents(seurat_object) <- "final_cluster_label"` or `Idents(seurat_object <- "disease_status")`. Next, differential expression is performed with the Seurat `FindMarkers()` function with a Wilcoxon Rank Sum test. Thresholds for logfc.threshold and min.pct are supplied by the user (in the form of slider bar inputs).

For local analysis, biological conditions can be made available for differential expression analysis with the additional_metadata_cols argument in the `export_shiny_object()` function.

Reclustering
Reclustering is performed based on the cluster identities supplied by the user. The input seurat object is subset for cells belonging to the cluster group(s) of interest before re-normalization and dimensionality reduction with the following commands:

```
new_object <- subset(seurat_object, idents = as.character(input$select_recluster))
new_object <- NormalizeData(new_object, normalization.method = "LogNormalize", scale.factor =
10000)
new_object <- FindVariableFeatures(new_object, selection.method = "vst", nfeatures = 2000)
all.genes <- rownames(new_object)
new_object <- ScaleData(new_object, features = all.genes)
new_object <- RunPCA(new_object, features = VariableFeatures(object = new_object))
new_object <- FindNeighbors(new_object, dims = 1:20)
new_object <- FindClusters(new_object, resolution = 0.5)
new_object <- RunUMAP(new_object, dims = 1:20)
new_object <- RunTSNE(new_object, dims = 1:20)
```