Case study: Monocle 3

R/

File	Functions
RcppExports.R	Needed for C++ integration
alignment.R	Data set alignment & batch effect correction
cell_data_set.R	Core classes (e.g. cell_data_set)
cluster_cells.R	Clustering cells in PCA, t-SNE, or UMAP space (e.g. louvain)
cluster_genes.R	Finding modules of co-regulated genes
expr_models.R	Regression modeling of gene expression data
find_markers.R	Finding biomarkers for cell types and clusters
generics.R	Implementations of standard BioC interfaces
graph_test.R	Tests for gene expression autocorrelation across cells
learn_graph.R	Inferring geometry of cell trajectories
load_cellranger_data.R	Importing 10X data
methods-cell_data_set.R	Core methods for cell_data_set objects
order_cells.R	Ordering cells in pseudotime
plotting.R	Many visualization functions
preprocess_cds.R	PCA, LSI
reduce_dimensions.R	Nonlinear embedding with t-SNE and UMAP
select_cells.R	Methods for selecting subsets of cells interactively
utils.R	Miscellaneous useful functions

Case study: Monocle 3

R/cell_data_set.R

```
new_cell_data_set <- function(expression_data,
                              cell_metadata = NULL,
                              gene_metadata = NULL) {

  assertthat::assert_that(class(expression_data) == "matrix" ||
                            is_sparse_matrix(expression_data),
                          msg = paste("Argument expression_data must be a",
                                      "matrix - either sparse from the",
                                      "Matrix package or dense"))
  …

  sce <- SingleCellExperiment(list(counts=methods::as(expression_data, "dgCMatrix")),
                              rowData = gene_metadata,
                              colData = cell_metadata)

  cds <- methods::new("cell_data_set",
             assays = SummarizedExperiment::Assays(
               list(counts=methods::as(expression_data, "dgCMatrix"))),
             colData = colData(sce),
             int_elementMetadata =int_elementMetadata(sce),
             int_colData = int_colData(sce),
             int_metadata = int_metadata(sce),
             metadata = metadata(sce),
             NAMES = NULL,
             elementMetadata = elementMetadata(sce)[,0],
             rowRanges = rowRanges(sce))

  metadata(cds)$cds_version <- Biobase::package.version("monocle3")
  clusters <- stats::setNames(SimpleList(), character(0))
  cds <- estimate_size_factors(cds)
  cds
}
```