hCoCena RStudio Docker Guide

Documentation

January 28, 2026

1 Overview

This guide describes how to build, update, and run the `hCoCena` package in a containerized RStudio environment. The setup is optimized for stability (fixed dependencies) and speed (smart caching).

2 File Setup

Ensure the following two files are located directly inside your `hCoCena-r-package` folder.

2.1 File 1: Dockerfile

This file defines the system environment. It uses `rocker/rstudio` as the base image and installs critical system libraries like `libzmq3-dev` for RCy3.

```
# 1. Base Image: RStudio Server (R Version 4.3.2)
FROM rocker/rstudio:4.3.2

# 2. System Dependencies
# Includes ’libzmq3-dev’ which is critical for RCy3/Cytoscape
RUN apt-get update && apt-get install -y \
    libcurl4-openssl-dev \
    libssl-dev \
    libxml2-dev \
    zlib1g-dev \
    libfontconfig1-dev \
    libfreetype6-dev \
    libharfbuzz-dev \
    libfribidi-dev \
    libpng-dev \
    libtiff5-dev \
    libjpeg-dev \
    libglpk-dev \
    libzmq3-dev \
    git \
    cmake \
    && rm -rf /var/lib/apt/lists/*

# 3. Set Working Directory inside the container
WORKDIR /home/rstudio/hcocena_project

# --- SMART CACHING LAYER ---
# Copy ONLY the dependency script first.
# Docker checks if this file changed. If not, it uses the cache.
COPY install_versioned_dependencies.R .

# 4. Install Dependencies
# Runs only if install_versioned_dependencies.R changes.
```

```
RUN Rscript install_versioned_dependencies.R

# --- PACKAGE INSTALLATION LAYER ---
# 5. Copy the actual package source code
COPY . .

# 6. Install the local hCoCena package
# Runs every time you change your R code.
RUN Rscript -e "remotes::install_local('.', dependencies = FALSE, upgrade = 'never')"

# Note: RStudio starts automatically via the base image.
```

Listing 1: Dockerfile

2.2 File 2: install_versioned_dependencies.R

This script installs R packages with specific versions and handles timeouts for large BioConductor downloads.

```r
# -----------------------------------------------------------
# 1. SETUP: Increase Timeout for BioConductor
# -----------------------------------------------------------
# Default is 60s, which is too short for large databases.
options(timeout = 12000)

if (!requireNamespace("remotes", quietly = TRUE)) install.packages("remotes")
library(remotes)

# Install BiocManager (Fixed version)
remotes::install_version("BiocManager", version = "1.30.22", upgrade = "never")

# -----------------------------------------------------------
# 2. CRITICAL FIXES (Archive & GitHub)
# -----------------------------------------------------------
print("--- Installing MCDA from Archive ---")
tryCatch(
  {
    remotes::install_url("https://cran.r-project.org/src/contrib/Archive/MCDA/MCDA_0.0.24.tar.gz", upgrade = "never")
  },
  error = function(e) {
    print("ERROR: MCDA installation failed!")
    quit(status = 1)
  }
)

print("--- Installing propr from GitHub ---")
remotes::install_github("tpq/propr", upgrade = "never")

# -----------------------------------------------------------
# 3. BIOCONDUCTOR DATABASES (First Priority)
# -----------------------------------------------------------
print("--- Installing heavy BioConductor Databases ---")
BiocManager::install(c("GO.db", "Biostrings", "DOSE", "AnnotationDbi", "HDO.db"), update=FALSE, ask=FALSE)

# -----------------------------------------------------------
# 4. BIOCONDUCTOR TOOLS (Second Priority)
# -----------------------------------------------------------
print("--- Installing BioConductor Tools ---")
BiocManager::install(c("clusterProfiler", "ComplexHeatmap", "leidenbase", "RCy3"), update=FALSE, ask=FALSE)

# -----------------------------------------------------------
# 5. STANDARD DEPENDENCIES (CRAN)
# -----------------------------------------------------------
print("--- Installing Standard Packages ---")

pkgs <- list(
  "Cairo" = "1.6-1",
  "circlize" = "0.4.15",
  "cowplot" = "1.1.1",
  "dendextend" = "1.17.1",
  "dplyr" = "1.1.4",
  "egg" = "0.4.5",
  "factoextra" = "1.0.7",
  "ggplot2" = "3.4.4",
  "ggsci" = "3.0.0",
  "gridExtra" = "2.3",
  "gtools" = "3.9.4",
  "Hmisc" = "5.1-1",
  "httr" = "1.4.7",
  "igraph" = "1.5.1",
  "jsonlite" = "1.8.7",
  "knitr" = "1.45",
  "leidenAlg" = "1.1.2",
  "magrittr" = "2.0.3",
  "networkD3" = "0.4",
  "openxlsx" = "4.2.5.2",
  "patchwork" = "1.1.3",
  "pheatmap" = "1.0.12",
  "plotly" = "4.10.3",
  "purrr" = "1.0.2",
  "RColorBrewer" = "1.1-3",
  "readr" = "2.1.4",
  "rlist" = "0.4.6.2",
  "scales" = "1.2.1",
  "stringi" = "1.8.1",
  "stringr" = "1.5.1",
  "tibble" = "3.2.1",
  "tidyselect" = "1.2.0"
)

for (pkg in names(pkgs)) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
      print(paste("Installing", pkg, "..."))
      remotes::install_version(pkg, version = pkgs[[pkg]], upgrade = "never")
  }
}

print("--- DONE: All dependencies installed ---")
```

Listing 2: install_versioned_dependencies.R

3 Building the Image

Open PowerShell in your `hCoCena-r-package` folder.

3.1 Scenario A: Smart Update (Routine)

Use this when you have only changed your R code (scripts/functions) but not the dependencies. Docker will use the cache for dependencies.

Time: Fast (∼ 30 seconds).

```
docker build -t mein-hcocena:latest .
```

3.2 Scenario B: From Scratch (Clean Build)

Use this when you want to rebuild everything from zero (e.g., on a new computer) or ensure a fresh environment.

Time: Slow (∼ 60 minutes due to downloads).

```
docker build --no-cache -t mein-hcocena:latest .
```

4 Running the Tool

Once the build is complete, use this command to start the server. This command mounts your local Windows folder into the container so you can save files.

Command:

```
docker run --rm -p 8787:8787 -e PASSWORD=test -v C:\Users\Thomas\hcocena:/home/
    rstudio/work mein-hcocena:latest
```

4.1 Access Instructions

1. Open your web browser.
2. Navigate to: `http://localhost:8787`
3. **Username:** `rstudio`
4. **Password:** `test`
5. To access your Windows files, look in the **Files** pane (bottom right) under the folder named **work**.