hanta™ R kit User Guide

(for Version 1.0)

I. Introduction

The hanta R kit is bioinformatic software for user-friendly analysis of sequencing data derived from Takara Bio platforms, such as the ICELL8® cx Single-Cell system and the ICELL8 Single-Cell system.

II. Before You Begin

A. Supported operating systems

The hanta software is designed to be installed on a user workstation and should work on any system that supports R (see below). Installation and functionality have been tested and supported for the following OSs:

- Windows 7
- MacOS X Sierra v10.12.6
- Linux Centos v6.9

B. Hardware requirements

hanta software with its dependencies is a lightweight program. It should work on any basic workstation (desktop or laptop) with > 2 GB of free disk space and a minimum of 8 GB RAM.

C. User account requirements

Administrative privileges are not required to install or run hanta software by default. If working in an environment where R is installed with IT restrictions, an administrator may need to install the necessary software dependences (Section I.D) and the hanta software.

D. Additional hardware and software dependencies and recommendations

- **Internet connectivity on the server**
- **R**

  R is a free, open-source software for statistical computing that provides support across a variety of operating systems. hanta software is designed to work within an R environment. More information on obtaining and installing R is available in Section IV.A.

- **RStudio Desktop**

  RStudio is a free, open-source program that provides graphical user interface (GUI) access to R. More information on obtaining and installing RStudio is available in Section IV.C.

- **devtools**

  devtools is a free, open-source R tool that enhances the development and installation of R packages; it's used to install the hanta software. More information on obtaining and installing RStudio is available in Section IV.D.

- **An open network port on the install machine**

  As the hanta software interface is accessed through a web GUI, a network port needs to be available on the computer it will be installed on. The port number is selected at random through the Rstudio `shiny runApp()` function until an open one is found. For more information about this assignment process, please see https://shiny.rstudio.com/reference/shiny/1.0.1/runApp.html.

  If running in an environment where the TCP/IP ports are locked down, please check with your local IT to ensure a port is available on the computer for hanta software to use.

- **Pandoc (optional)**

  Pandoc is another R utility that is installed natively with RStudio Desktop. For advanced users that wish to forgo RStudio and run R from the command line, the Pandoc Software Package must be downloaded, installed, and placed in the PATH.

E. Required input files

hanta software requires one of the two following file options as input:

- A cluster data R-object from mappa™ Analysis Pipeline (mappa). The advantage to this input is that the quality control and clustering modules have been pre-calculated, resulting in faster upload speeds.

- Raw gene expression matrix and metadata files. These allow the user to interactively run the data through quality control filtering and clustering. More information on this process can be found in Section VII.

III. Software overview

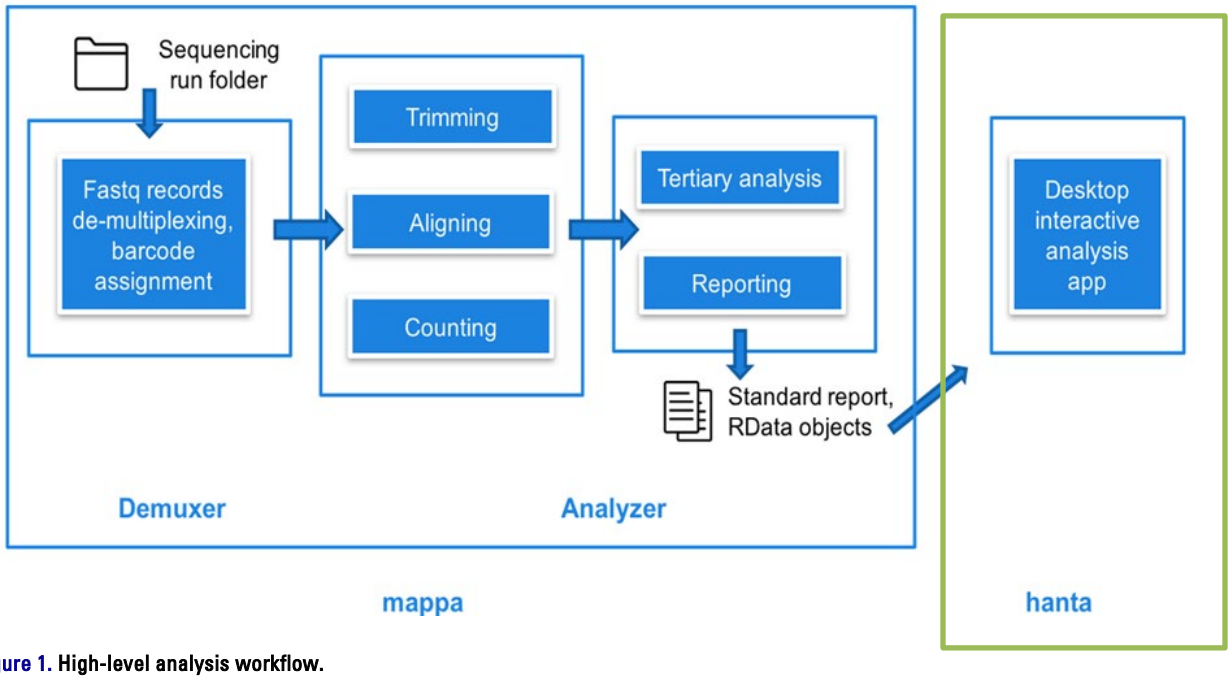


Figure 1. High-level analysis workflow.

Figure 1 (above) depicts the high-level workflow of the analysis provided by mappa and how its output can be carried over to the hanta R kit. For more information about mappa, see the mappa Analysis Pipeline User Guide at takarabio.com.

Once the hanta software and required dependencies are installed, the standard analysis can be run in an interactive RStudio session or on the command line.

IV. Installation and configuration requirements

To obtain the hanta software, please visit the ICELL8 software portal at takarabio.com/ICELL8-software.

A. Install R

R and many of the contributed packages are available on the Comprehensive R Archive Network (CRAN). If R is not installed on your system, please download and install R version 3.5.0 or higher from cran.cnr.berkeley.edu, or the CRAN mirror of your choice.

For detailed instructions about how to install R on Windows, see Appendix D.

B. Install platform-specific tools

Installation of hanta software on Windows or Macintosh workstations requires additional third-party software be installed prior to Step C.

1. Windows

On Windows, R requires Rtools to build and install packages from source file. Download Rtools from cran.cnr.berkeley.edu/bin/windows/Rtools. During installation, ensure that Rtools is included in the system PATH.

For a detailed walkthrough of how to install Rtools, see Appendix E.

> **NOTE**: Rtools must be installed in a file path with directory names which do not include spaces. (i.e., it cannot be installed in `C:\Program Files\`, but could be installed in `C:\Program\`). Installing it in a file path with spaces in the directory names will cause the hanta software installation to fail.
>
> If Rtools is installed in such a location on the target computer, please uninstall Rtools and re-install in a folder with a path that conforms to these requirements.

2. MacOSX

R version 3.5.x running on MacOSX requires installation of `clang-6.0.0.pkg` and `gfortran-6.1.pkg`. R version 3.6.x requires `clang-7.0.0.pkg`. These can be downloaded from cran.cnr.berkeley.edu/bin/macosx/tools.

C. Install RStudio

If RStudio is not installed on your system, please download and install the RStudio Desktop (Open Source License) version for your Operating System from rstudio.com.

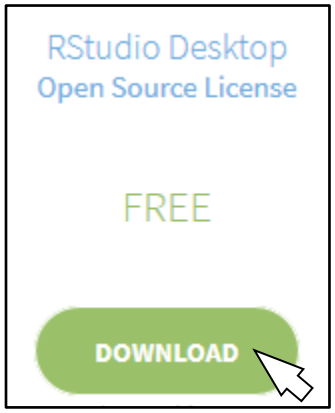


Figure 2. rstudio.com screenshot of the package to download.

For a detailed walkthrough of how to install RStudio on Windows, see Appendix F.

D. Install devtools

The hanta pipeline requires devtools version 2.0.1 or later be installed on the computer prior to the hanta software installation.

1. If devtools is already installed on the computer, verify that the version is 2.0.1 or later by running the following command from the Console prompt of RStudio:

```
packageVersion("devtools")
```

```
> packageVersion("devtools")
[1] '2.0.1'
```

Figure 3. Example devtools version check in RStudio.

2. To update the devtools version or for a new install, enter the following command into the Console window of RStudio:

```
install.packages("devtools")
```

or follow the detailed walkthrough to install on Windows via the GUI in Appendix G.

E. (Optional) Install Pandoc

Instructions for downloading and installing Pandoc can be found at pandoc.org.

F. Install hanta software

Once the prerequisites are installed, hanta software can be installed with the following command.

```
devtools::install_github("takarabiousa/hanta", auth_token = "<AUTHCODE>")
```

where <AUTHCODE> will be a unique authorization token provided via email.

```
Downloading GitHub repo takarabiousa/hanta@master
Installing 116 packages: acepack, annotate, AnnotationDbi, base64e
```

Figure 4. Example of the text displayed when the hanta software installation starts.

To obtain the authorization code, please register at the hanta software page at takarabio.com/ICELL8-software.

For first time users, the installation process may take 10–20 minutes, as many dependencies are automatically downloaded and installed. The installation may also prompt the user to accept downloading and installing certain packages from source. Answer yes to any such prompts.

G. Upgrading hanta software

The procedure to upgrade hanta software is to the same as the procedure for doing an installation (Section IV.F, above).

It may be that during the upgrade, the script will notice updates to the R dependencies installed along with hanta software. If that occurs, it is recommended to select whatever the 'All' value is (33 in the Figure 5 example, below).

```
Downloading GitHub repo takarabiousa/hanta@master
These packages have more recent versions available.
Which would you like to update?

 1:   assertthat   (0.2.0        -> 0.2.1       ) [CRAN]
 2:   callr        (3.1.1        -> 3.2.0       ) [CRAN]
 3:   cli          (1.0.1        -> 1.1.0       ) [CRAN]
 4:   colorspace   (1.4-0        -> 1.4-1       ) [CRAN]
 5:   formatR      (1.5          -> 1.6         ) [CRAN]
 6:   glue         (1.3.0        -> 1.3.1       ) [CRAN]
 7:   highr        (0.7          -> 0.8         ) [CRAN]
 8:   httpuv       (1.4.5.1      -> 1.5.0       ) [CRAN]
 9:   kableExtra   (1.0.1        -> 1.1.0       ) [CRAN]
10:   knitr        (1.21         -> 1.22        ) [CRAN]
11:   lazyeval     (0.2.1        -> 0.2.2       ) [CRAN]
12:   mvtnorm      (1.0-8        -> 1.0-10      ) [CRAN]
13:   openssl      (1.2.1        -> 1.2.2       ) [CRAN]
14:   processx     (3.2.1        -> 3.3.0       ) [CRAN]
15:   purrr        (0.3.0        -> 0.3.2       ) [CRAN]
16:   Rcpp         (1.0.0        -> 1.0.1       ) [CRAN]
17:   RCurl        (1.95-4.11    -> 1.95-4.12   ) [CRAN]
18:   registry     (0.5          -> 0.5-1       ) [CRAN]
19:   rlang        (0.3.1        -> 0.3.2       ) [CRAN]
20:   rmarkdown    (1.11         -> 1.12        ) [CRAN]
21:   robustbase   (0.93-3       -> 0.93-4      ) [CRAN]
22:   rstudioapi   (0.9.0        -> 0.10        ) [CRAN]
23:   stringi      (1.3.1        -> 1.4.3       ) [CRAN]
24:   sys          (2.1          -> 3.1         ) [CRAN]
25:   tibble       (2.0.1        -> 2.1.1       ) [CRAN]
26:   tidyr        (0.8.2        -> 0.8.3       ) [CRAN]
27:   tinytex      (0.10         -> 0.11        ) [CRAN]
28:   xfun         (0.4          -> 0.5         ) [CRAN]
29:   XML          (3.98-1.17    -> 3.98-1.19   ) [CRAN]
30:   NMF          (9e70be3ce... -> f1bc224eb...) [GitHub]
31:   Annotatio... (1.42.1       -> ce191b08c...) [GitHub]
32:   CRAN packages only
33:   All
34:   None
Enter one or more numbers separated by spaces, or an empty line to
cancel
1:
```

Figure 5. Example of the hanta software upgrade process detecting newer R dependency packages.

If an error is thrown indicating Rstudio could not remove a prior package installation, please see Appendix A for one potential fix.

H. Uninstalling hanta software

To uninstall hanta software, run the following command at the Rstudio prompt:

```
remove.packages("hanta")
```

V. hanta standard analysis

Once the hanta software and required dependencies are installed, the standard analysis can be run in an interactive RStudio session or directly on the command line.

A. Overview

The hanta standard analysis tool is designed to provide users with the ability to control parameters to summarize their single-cell sequencing data. Users can select options to perform QC filtering, normalization, and transformation, and to generate an html report covering hanta software's correlation, clustering, and gene expression modules. A gene matrix and stats/metadata file are required for input.

NOTE: Sample data to test either the basic or advanced usage commands can be downloaded by typing the following command in the RStudio console:

```
hanta.example_data("hanta_test", "<path>")
```

where `<path>` is replaced with the full path on your desktop where you want the sample data to be copied to. The file `hanta_test.zip` file will download to the specified `<path>` folder.

E.g.,

```
hanta.example_data("hanta_test", "C:/mappa_data")
```

The file will need to be unzipped before use.

B. Basic usage

The only option required to run hanta software's standard analysis function is `mappa_data_loc`, which is the location of the mappa pipeline output folder. This parameter directs the program to the location on the workstation where the `genematrix.csv` and `stats.csv` files are stored.

The following steps are the minimum required to run the standard analysis with default settings.

1. Run the RStudio program to bring up the RStudio user interface.
2. In the Console window command-line, load the hanta library with the command:

   ```
   library(hanta)
   ```

3. Run the basic analysis with the command:

   ```
   hanta.analysis(mappa_data_loc = "%MAPPA_PATH%/mappa_data")
   ```

   Replace `%MAPPA_PATH%` with the local location of the folder output from the mappa pipeline.

   NOTE: Dividers between directory names should be a forward slash on Windows, not the Windows-usual backslash, i.e., `C:\mappa_data` will generate an error.

   After executing the analysis command, the script may take several minutes to complete.

   Example:

   If the data files are located in `C:\mappa_data`

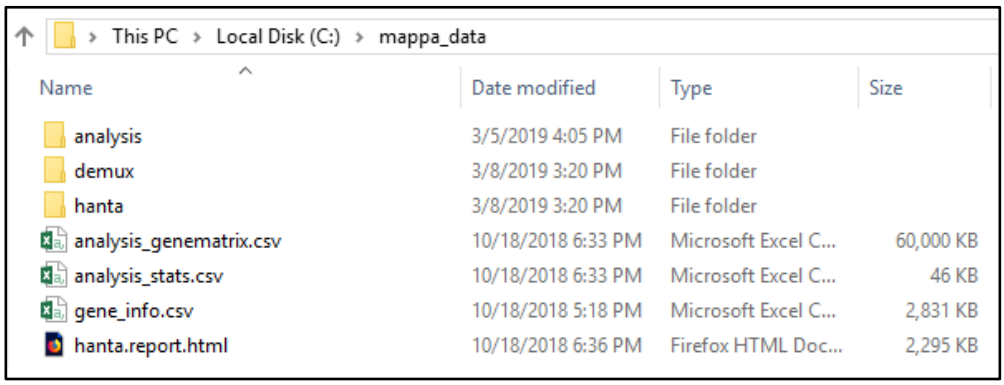
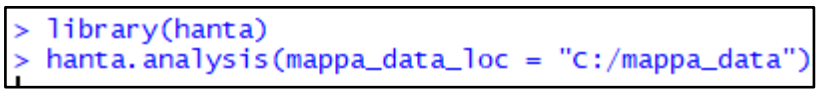


Figure 6. Example `mappa_data` directory contents.

At the RStudio prompt, type:

```
hanta.analysis(mappa_data_loc = "C:/mappa_data")
```



C. Advanced options

The standard analysis can be further optimized by selecting parameters for advanced customization. To add an option, include it within the `hanta.analysis()` call.

The following examples illustrate customizing the options for the analysis run. Additional configuration parameters can be found in Appendix B.

Example:

Determine what parameters to apply during standard analysis. The advanced parameters listed below were identified for this run.

a. Customize normalization options (`gm_norm = "cpm" & gm_norm_scale = 10000`)
b. Perform natural logarithm transformation (`gm_log_base = "ln"`)
c. Set the grouping variable to "Sample" to provide analyses by cell type (`grouping_var = "Sample"`)
d. Set the `verbose` option to TRUE to get progress readouts (`verbose = TRUE`)
e. Output the text to a specific directory on the computer rather than the default (`output_dir = "%OUTPUT_PATH%/output"`)

1. Load hanta software into RStudio

   # load hanta library

   ```
   library(hanta)
   ```

2. Run the analysis commands from the RStudio Console window command line:

   # run analysis on chosen advanced settings and values

   ```
   hanta.analysis(mappa_data_loc = "%MAPPA_PATH%/mappa_results",
                  gm_norm = TRUE,
                  gm_norm_method = "cpm",
                  gm_norm_scale = 10000,
                  gm_log = TRUE,
                  gm_log_base = "ln",
                  grouping_var = "Sample",
                  output_dir = "%OUTPUT_PATH%/output",
                  verbose = TRUE)
   ```

NOTE: Make sure to specify the `%MAPPA_PATH%` and `%OUTPUT_PATH%` directory values to match the file location on your system.

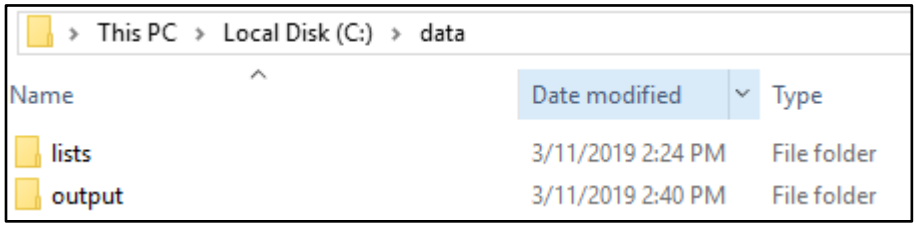
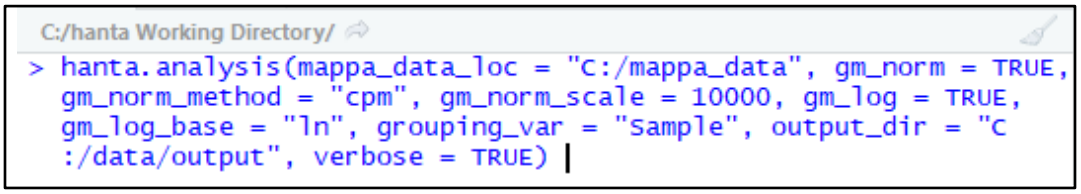


Figure 7. `data` **directory content for the example command, below.**

Example:

```
hanta.analysis(mappa_data_loc = "C:/mappa_data", gm_norm = TRUE,
  gm_norm_method = "cpm", gm_norm_scale = 10000, gm_log = TRUE,
  gm_log_base = "ln", grouping_var = "Sample", output_dir =
  "C:/data/output", verbose = TRUE)
```



VI. hanta output

Unless the `output_dir` option is specified (see Section V.C, above, for an example of doing this), the analysis output from hanta software is directed to the RStudio current working directory into a sub-directory called `hanta_output`.

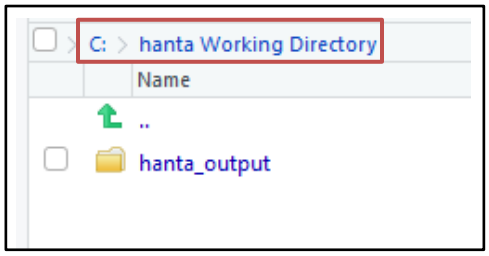


Figure 8. **Current working directory example in RStudio and the** `hanta_output` **default output directory.**

NOTE: See Appendix F.B for steps to configure the RStudio current working directory.

The primary output file is a report, entitled `hanta.report.html`, which contains an overview of the analysis. The output also includes several figures, plots, and R objects which can be reloaded for further analysis.

For a detailed breakdown of hanta software output files, see Appendix C.

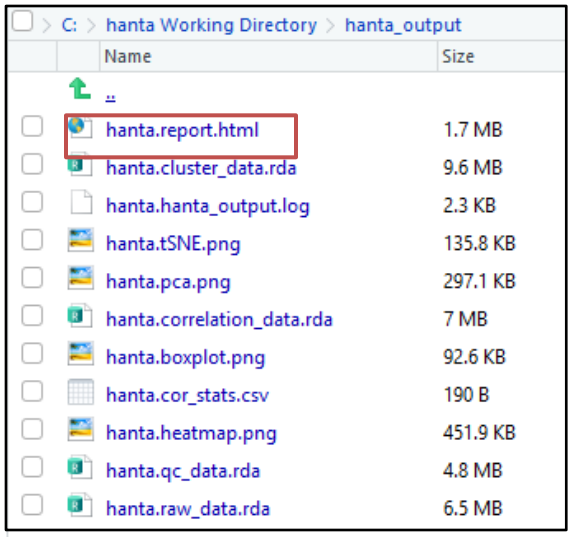


Figure 9. **Primary output file, `hanta.report.html`, in the `hanta_output` default output directory in RStudio file browser window.**

VII. hanta interactive application

A. Getting started

Once installation is complete, the hanta interactive application can be launched with the following command in an open RStudio session.

```
hanta()
```

```
> hanta()
```

Entering this command will launch the default browser on your computer and create a new instance of the hanta UI, running through the localhost of your computer (IP address 127.0.0.1) and a randomly assigned, available TCP/IP port (in Figure 10, below, the port chosen is 6863).

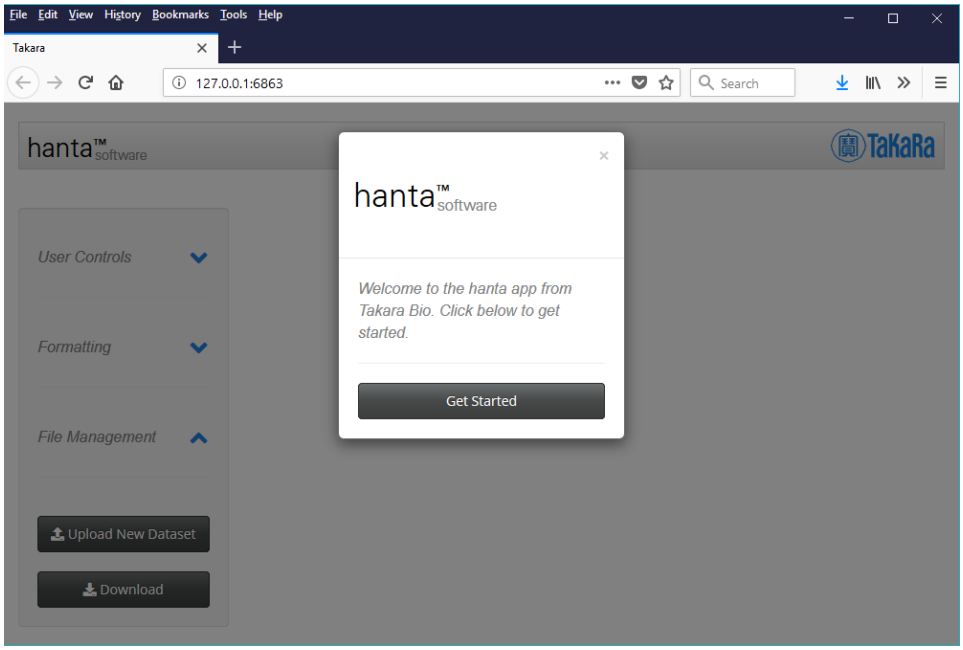


Figure 10. **Initial screen of the hanta interactive application in the web browser.**

B. Upload data

Click [Get Started] to start the process. The *Select Input Data* window will pop up.

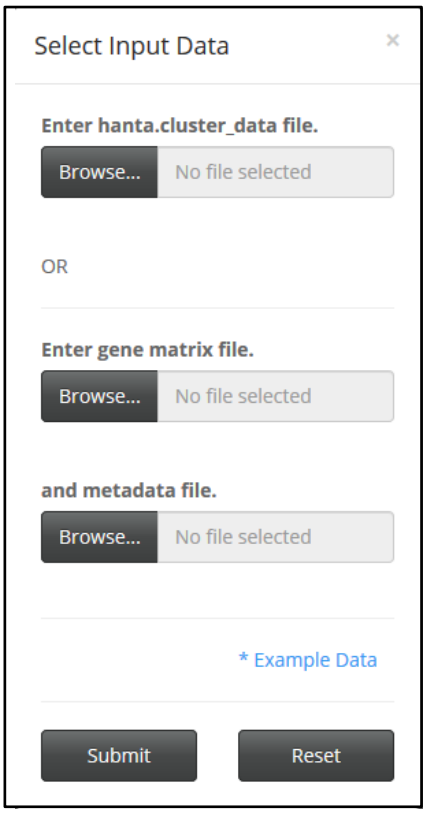


Figure 11. The *Select Input Data* browser pop-up window.

> **NOTE**: The Select Input Data menu can also be accessed with the [Upload New Dataset] button from the **File Management** menu in an established hanta session. See Figure 41 in Section VII.F for where to locate the button.

The *Select Input Data* window allows the user to enter data in one of two ways.

1. Through a cluster data R object, which is output through mappa Analysis Pipeline.

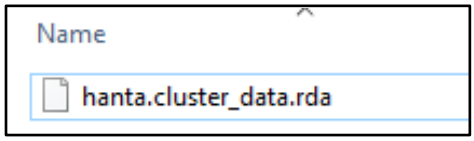


Figure 12. Example cluster data R object file generated by the mappa software.

2. Raw gene expression matrix and metadata files.

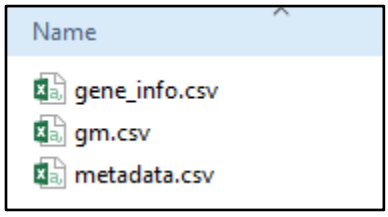


Figure 13. Example data CSV files.

The primary difference between the two options is that the cluster data object has already been run through the quality control and clustering modules in the mappa software. If a cluster data r object (Figure 12) is entered, skip to Section VII.E (below).

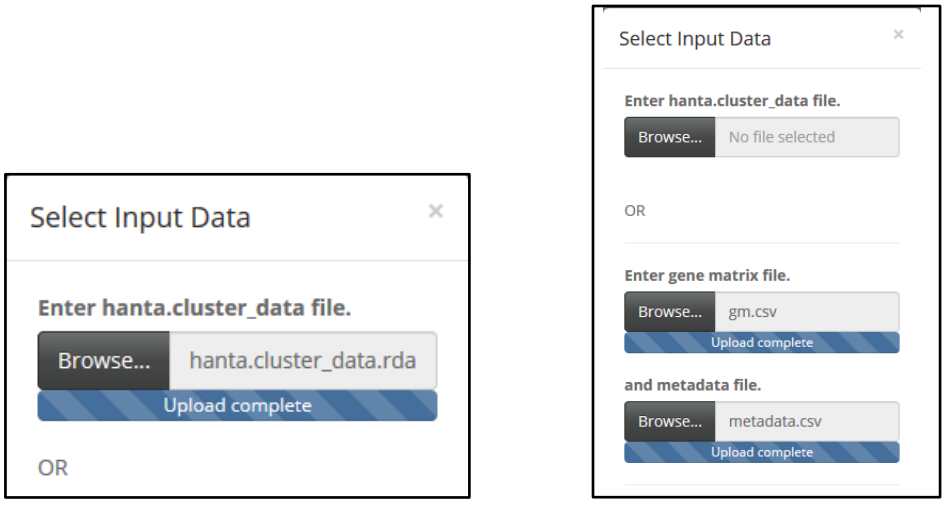


Figure 14. Selecting the example mappa output cluster data R object (left) or CSV gene matrix and metadata files (right) for input.

Entering the raw gene expression matrix and metadata file allows the user to interactively run the data through quality control filtering and clustering, described in the rest of this section.

Once the input data source is selected, hit [Submit] to continue.

NOTE: The `Example Data` link in Figures 11 and 15 downloads data from the study "Massively parallel nanowell-based single-cell gene expression profiling" (Goldstein et al. 2017).

The screenshots in the rest of this section are based on that sample data.

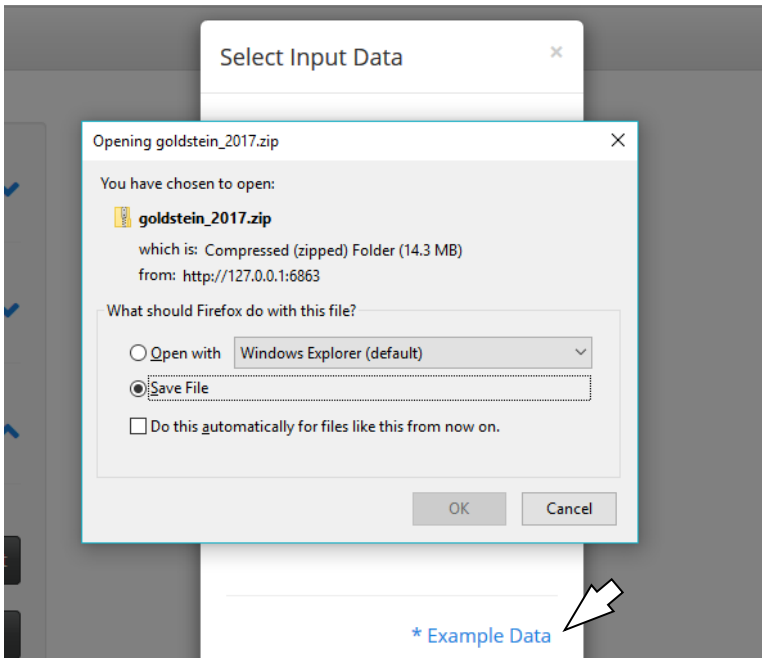


Figure 15. Download prompt for the sample Goldstein *et al.* 2017 sample data, from the `Example Data` link of the hanta UI.

C. Format of the gene expression matrix and metadata files

- The gene expression matrix file (`gm.csv` in Figure 13, above) must be in comma-separated values (CSV) format containing columns of unique sample identifiers, with rows of gene names. Each entry in the matrix is an expression value representing the expression of gene (i) for sample (j). The expression data may be raw count data or pre-normalized/transformed data.
- The metadata file (`metadata.csv` in Figure 13, above) must be in CSV format with one column containing the unique sample identifiers used in the gene expression matrix and any number of subsequent columns with metadata for each sample (i.e., cell type, gene counts, read depth, mitochondrial %, etc.)

D. Run Quality Control and Clustering modules on raw datasets

Entering the raw gene expression matrix and metadata files prompts the user to enter options for hanta software's *Quality Control* module.

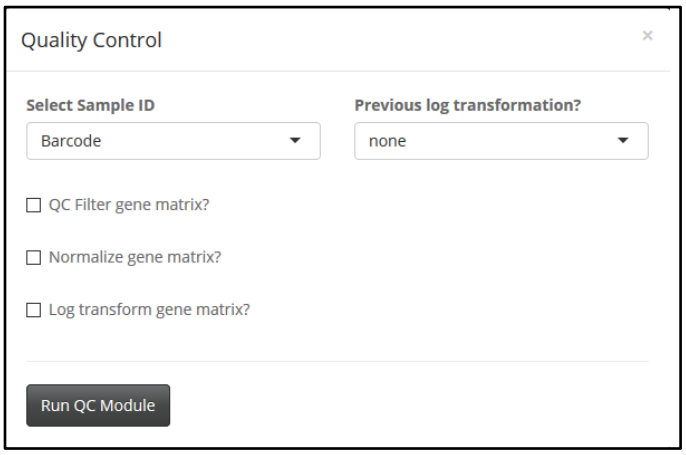


Figure 16. Default *Quality Control* menu.

1. Click the check boxes next to the section questions to expand to the view seen in Figure 17. (below).

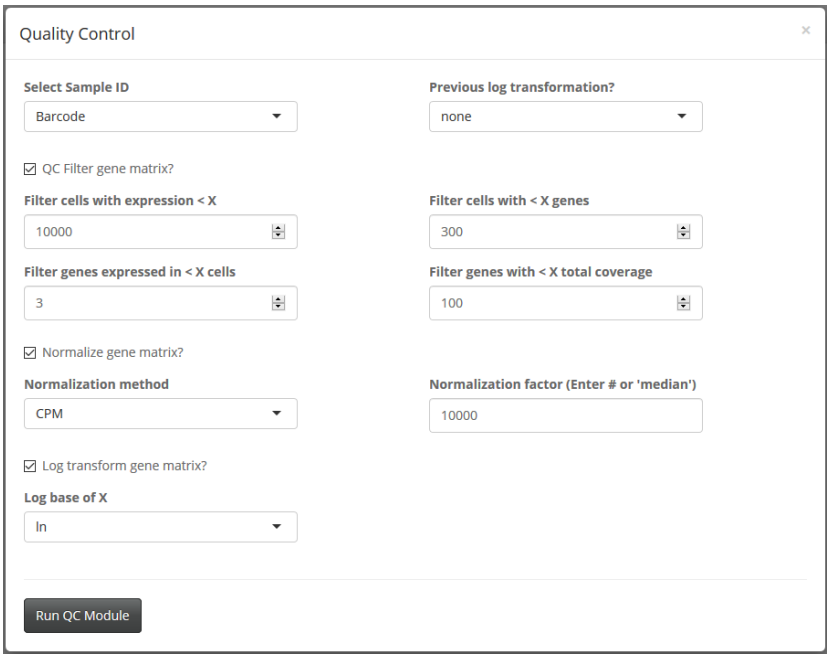


Figure 17. *Quality Control* menu, expanded.

a. **Select Sample ID**

Select the column header for the Sample ID from the metadata file that matches the Sample IDs used in the gene expression matrix. This option will be pre-populated with column headers from the metadata file.

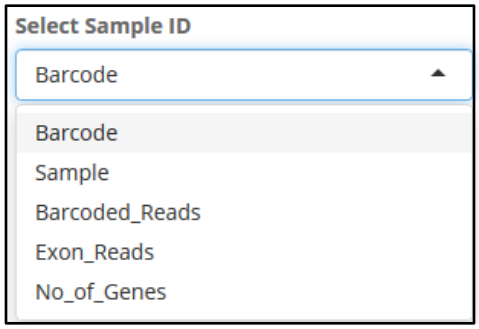


Figure 18. Expanded Select Sample ID drop-down menu.

b. **Previous log transformation**?

If the data has been previously log-transformed, please enter the log-base used from (ln, 2, or 10).

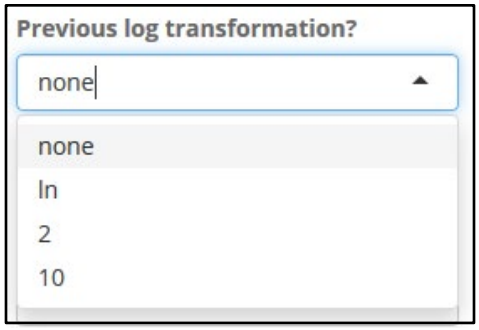


Figure 19. Expanded Previous log transformation drop-down menu.

c. **QC Filter gene matrix**?

The user may select how to filter non-informative cells and genes from the gene expression matrix.

d. **Normalize gene matrix?**

The available normalization methods include Counts Per Million (CPM), Transcripts Per Kilobase Million (TPM), and Reads Per Kilobase Million (RPKM).

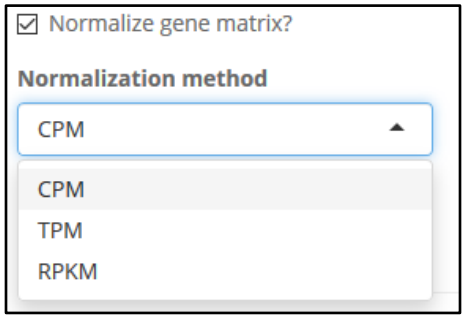


Figure 20. Expanded Normalize gene matrix drop-down menu.

To normalize by 'median cell coverage', select 'CPM' from the Normalization method" drop box and type 'median' into the "Normalization factor" input box (Figure 21).

Figure 21. Parameters to normalize by median cell coverage.

e. **Log transform gene matrix?**

To log transform the data, the available options are natural log, Base 2, and Base 10.

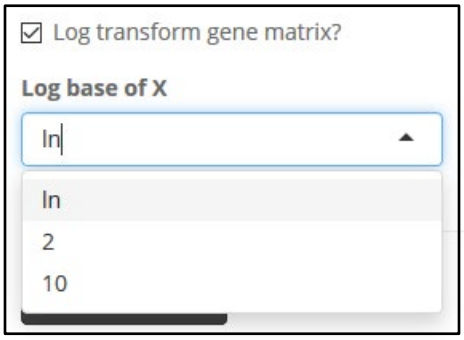


Figure 22. Expanded Log transform gene matrix drop-down menu.

2. When all desired parameters are populated, click [Run QC Module].
3. A window will pop up prompting to enter options for the *Cluster Analysis* module. In this example, cluster analysis based on the 500 most variable genes is selected.

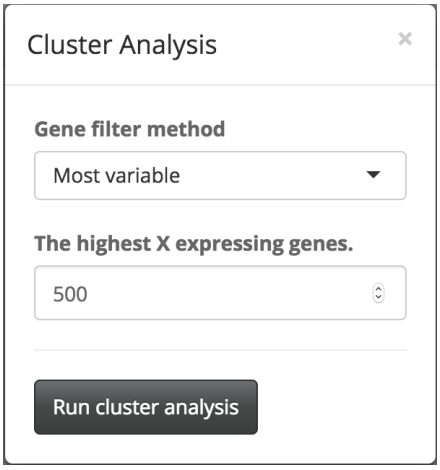


Figure 23. Cluster Analysis menu.

4. Click [Run cluster analysis], and data transformation will begin. A status pop-up similar to those in Figure 24 will display on the bottom right-hand corner of the browser window.

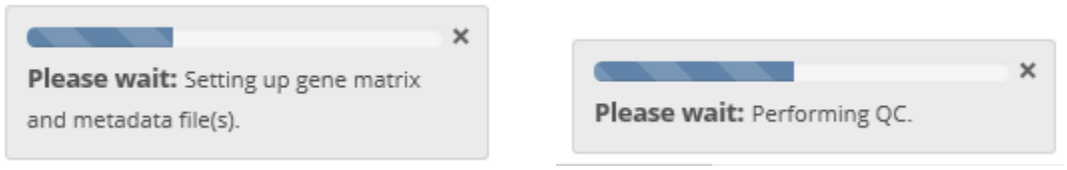


Figure 24. Two stages of the status pop-up while the cluster analysis is running.

5. After running the cluster analysis, the plot will be rendered in the center of the app.

E. Explore the data

The baseline plot is displayed in grayscale by default but can be modified with the **User Controls** and **Formatting** menu options in the sidebar panel to the left of the screen.

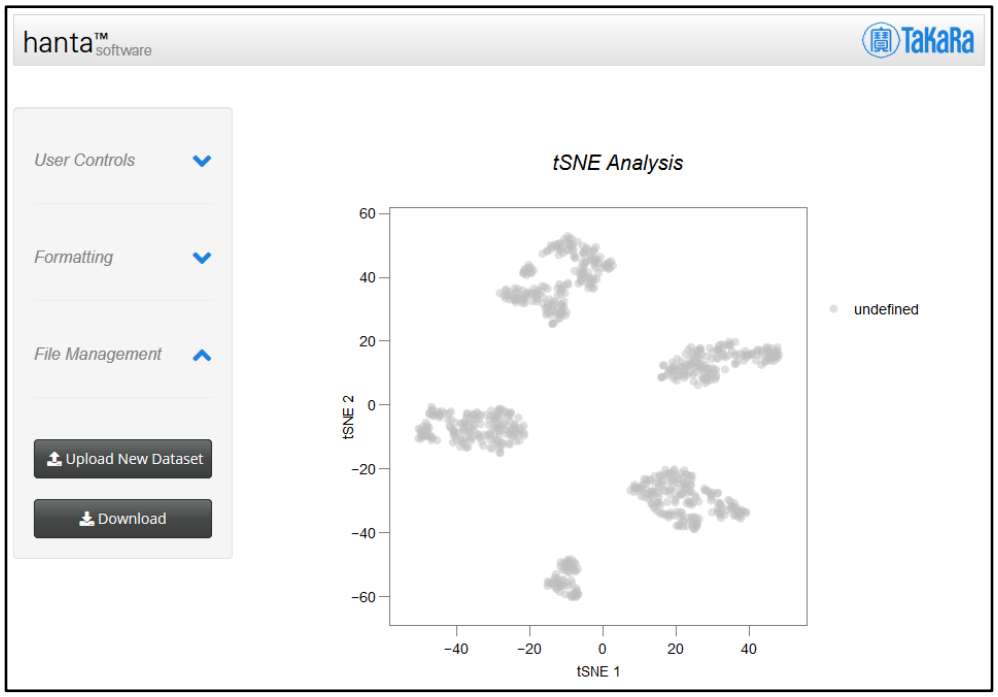


Figure 25. Baseline tSNE analysis plot.

1. User Controls

   a. To highlight cells by cell type, open the **User Controls** accordion menu and select an option from the "Select Group(s)" drop-down box. This field is pre-populated with all column headers from the metadata file. Selecting different metadata features allows the user to highlight the cells by any desired method.

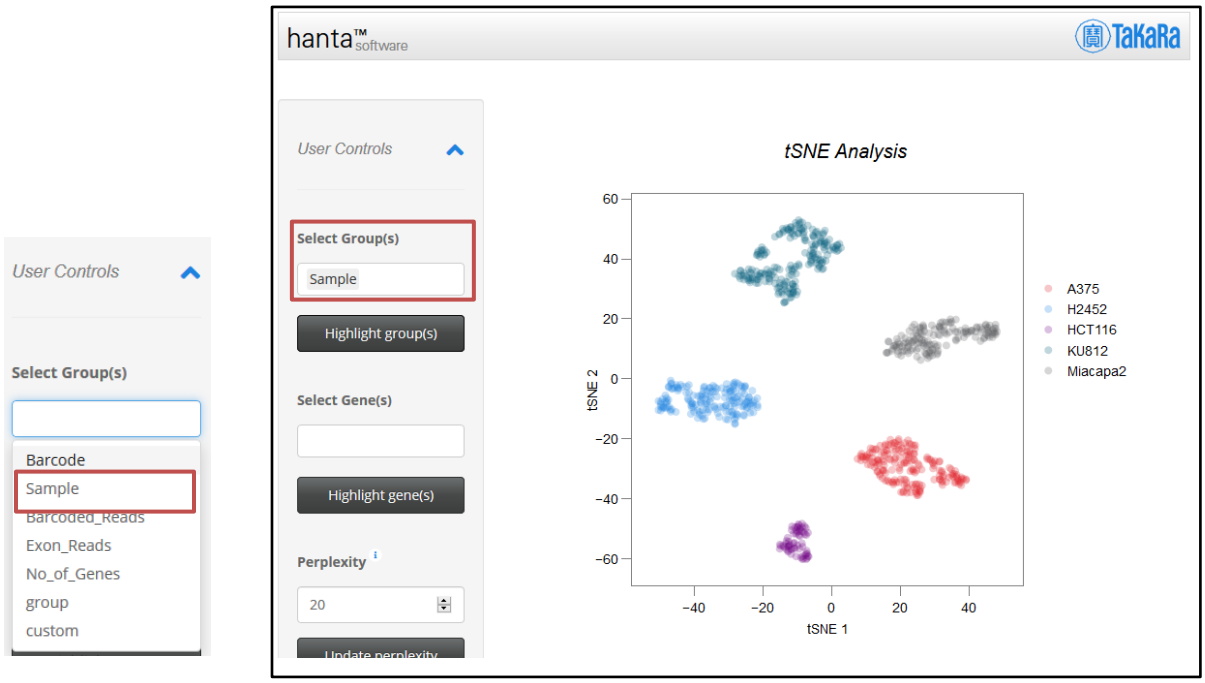


Figure 26. (left) Example "Select Group(s)" drop-down menu, (right) the resulting tSNE analysis plot with cells highlighted by cell sample type.

b. Another method to highlight cells is by expression levels for genes. Entering one or multiple genes into the "Select Gene(s)" field plots the average expression across the panel for each cell and renders the expression into the plot. In Figure 27 (below), the plot highlights ENSG00000261857, a single marker for the A375 cell type.

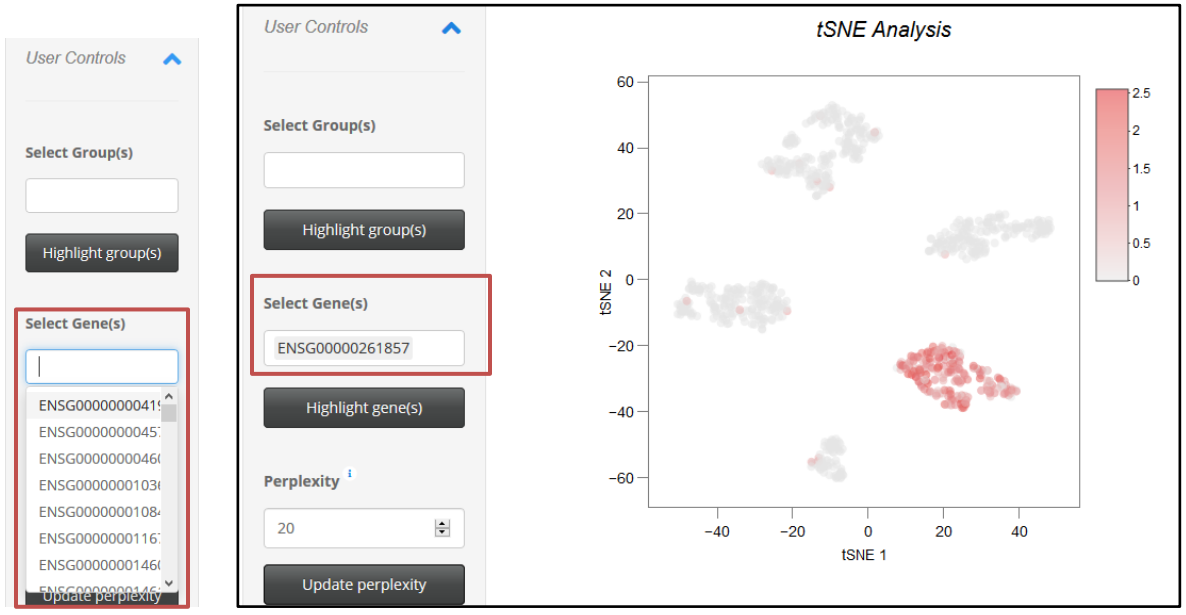


Figure 27. (left) Example "Select Gene(s)" drop-down menu, (right) the resulting tSNE analysis plot with cells highlighted for gene ENSG00000261857.

c. The "perplexity" parameter can also be configured. A feature of the tSNE calculation that broadly serves as an estimate of the cluster size(s) within the data, high perplexity parameters will define large, global structures within the dataset, while smaller perplexities will identify small, local structures.

NOTE: Perplexity defaults have been optimized for general use cases of the ICELL8 cx Single-Cell and ICELL8 Single-Cell systems. These values are different from the standard defaults in the Rtsne package and may need to be reoptimized for unique applications.

For more information, please refer to https://github.com/jkrijthe/Rtsne, https://distill.pub/2016/misread-tsne/, and https://cran.r-project.org/web/packages/Rtsne/Rtsne.pdf.

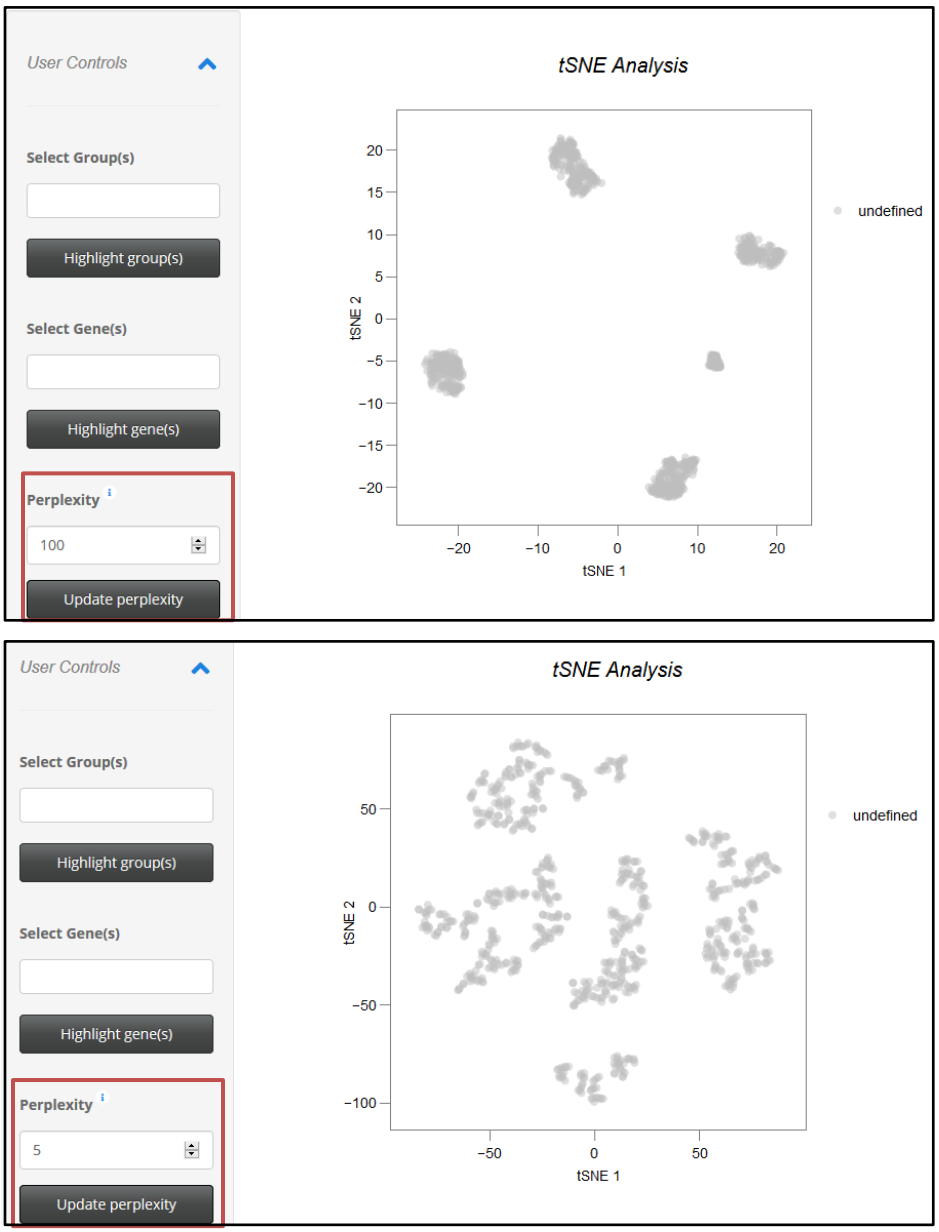


Figure 28. tSNE analysis plot with cells clustered by high (100) perplexity (above) and low (5) perplexity (below).

2. Formatting

In the "Formatting" tab in the sidebar panel, the marker size and transparency can be changed. These can be used to visualize the data to the user's preferences but are also useful for identifying individual cells within larger clusters.

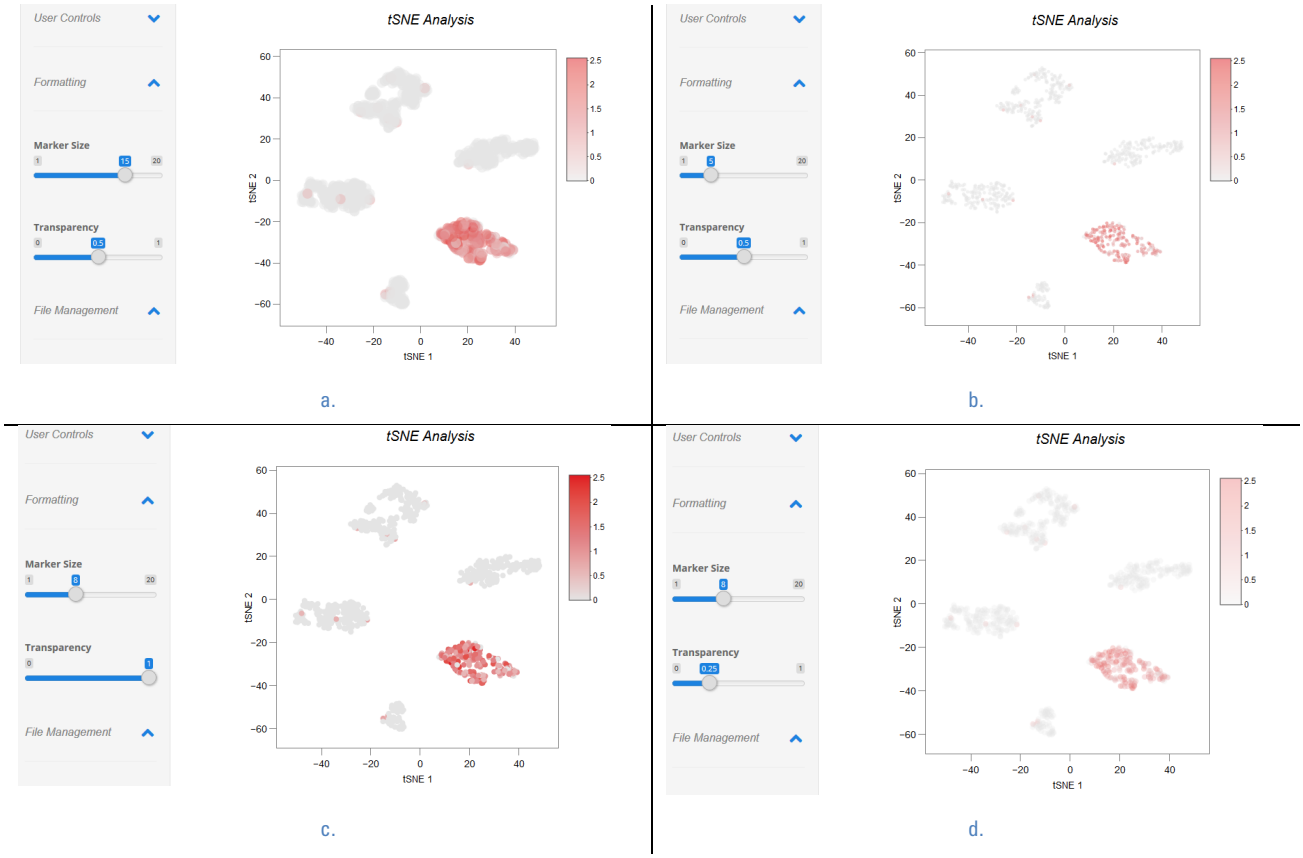


Figure 29. **Illustration of different marker sizes and transparency selections in the "Formatting" option applied to the same data plot. (Top) (a) larger and (b) smaller marker size. (Bottom) (c) less and (d) more transparency.**

3. Floating menu

To the right of the chart, there is a menu of icons that only displays when hovered over with the mouse cursor.

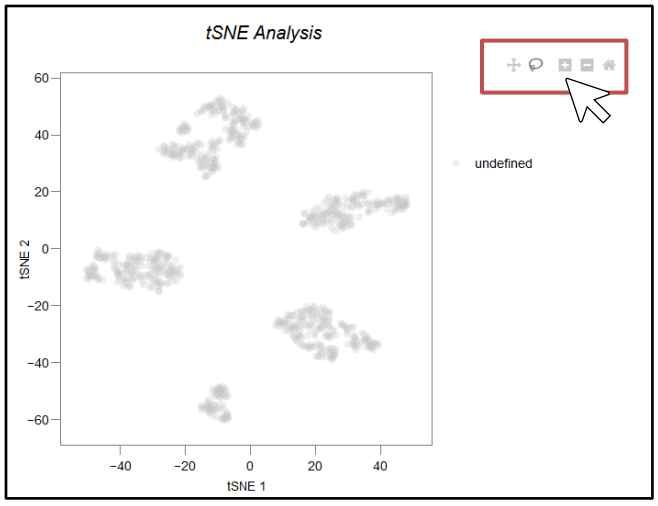


Figure 30. **Location of the floating menu icons to the right of the tSNE chart.**

a. Pan and Reset axes

Figure 31. Identification of the Pan and Reset axes icons in the floating menu.

i. The Pan function can be used to move the scatter plot within the frame of the chart axes, changing not just what plots are visible, but also the labels on the X- and Y-axis.

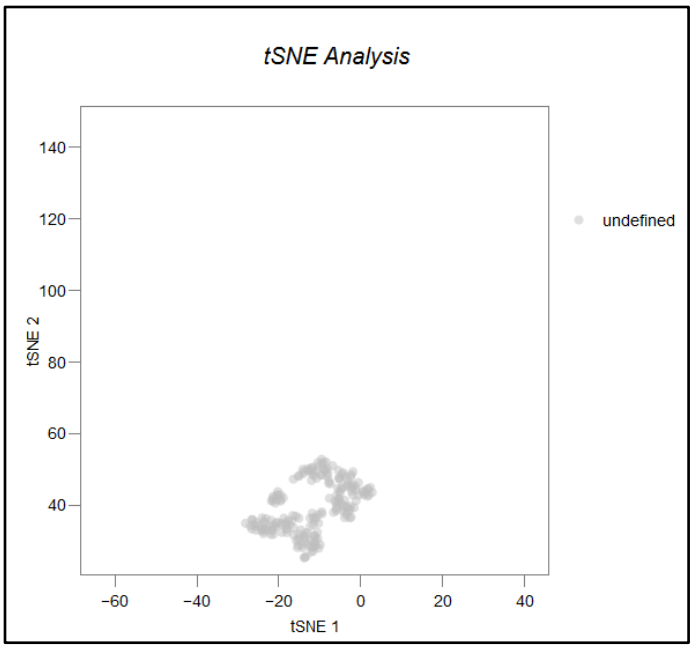


Figure 32. Example after using the Pan function to move the plots down the page, increasing the values of the Y-axis compared to the default in Figure 30.

ii. The Reset axes will return the plot to the default view (Figure 30) after using the Pan, Zoom in, and/or Zoom out functions.

b. Zoom in and Zoom out

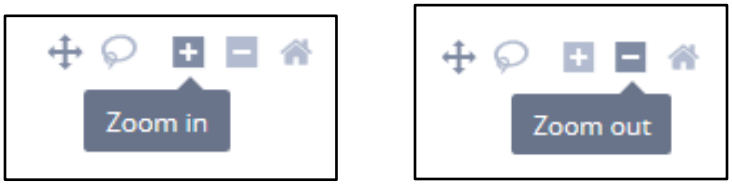


Figure 33. Identification of the Zoom in and Zoom out icons in the floating menu.

The Zoom in and Zoom out buttons can be used to either enlarge or shrink the plots within the chart, decreasing or increasing the scale of the axes (respectively).

c. Lasso select

Figure 34. Identification of the Lasso Select icon in the floating menu.

The Lasso Select feature can be used to select, group, and label cells in a custom manner.

i. Click the [Lasso Select] icon.

ii. Left-mouse click in the plot area and, while holding the mouse button down, use the mouse cursor to draw around the cells of interest. The line will automatically adjust its shape based on the movement of the mouse cursor.

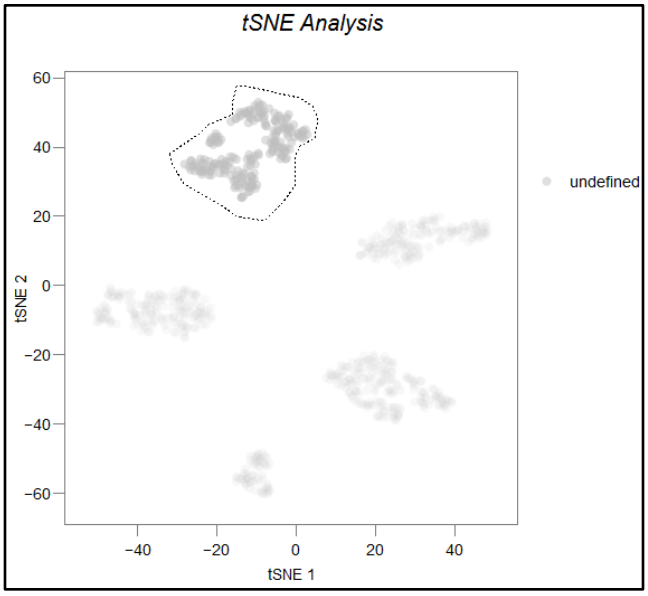


Figure 35. Lassoing a cell cluster of interest.

iii. Stop pressing on the left-mouse button, and the *Custom Selection* window will pop up.

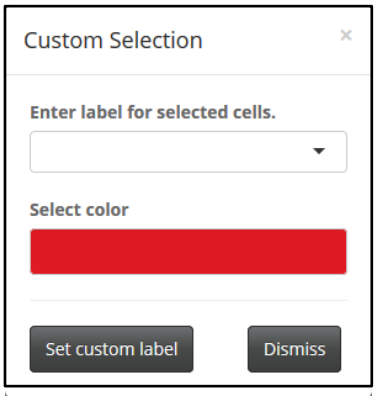


Figure 36. Default Custom selection pop-up window.

1. **Enter label for selected cells** – (Optional) Type in text that will identify the cluster in the legend on the right-hand side of the chart.

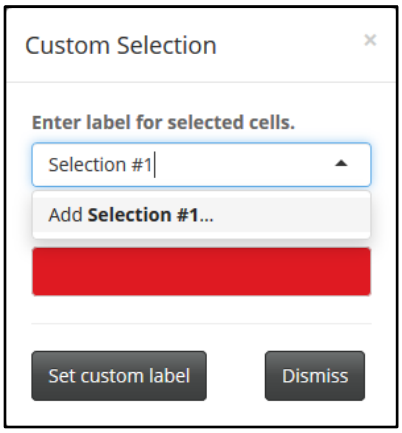


Figure 37. Typing in a custom label name for the lassoed cells on the chart.

2. **Select color** – (Optional) Click on the color bar to expand out to a color selector gradient. Macro changes can be made on the vertical rainbow bar, while finer gradients can be selected by moving the dot around the larger square of color shades on the left.

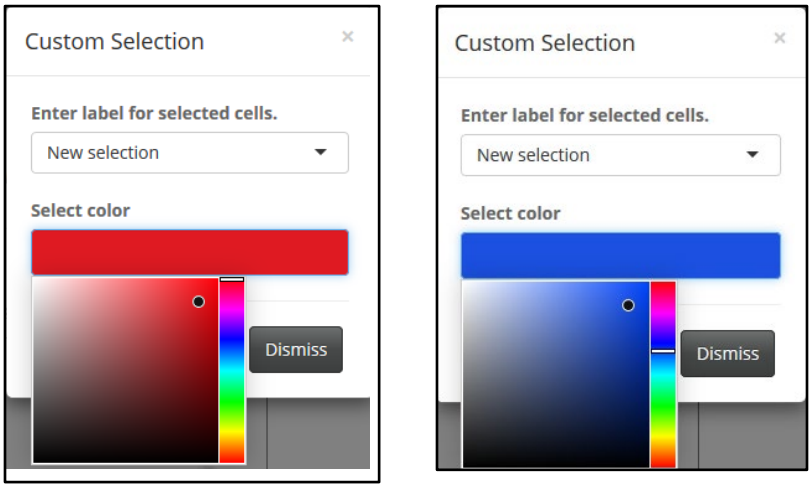


Figure 38. Examples of two different color selections using the macro and finer gradients fields.

iv. Once the options are selected, hit the [Set custom label] button to apply them.

To quit without applying the customization, press the [Dismiss] button.

v. If the customizations are set, the *Custom Selection* pop-up will disappear, and the chart will reflect the changes made.

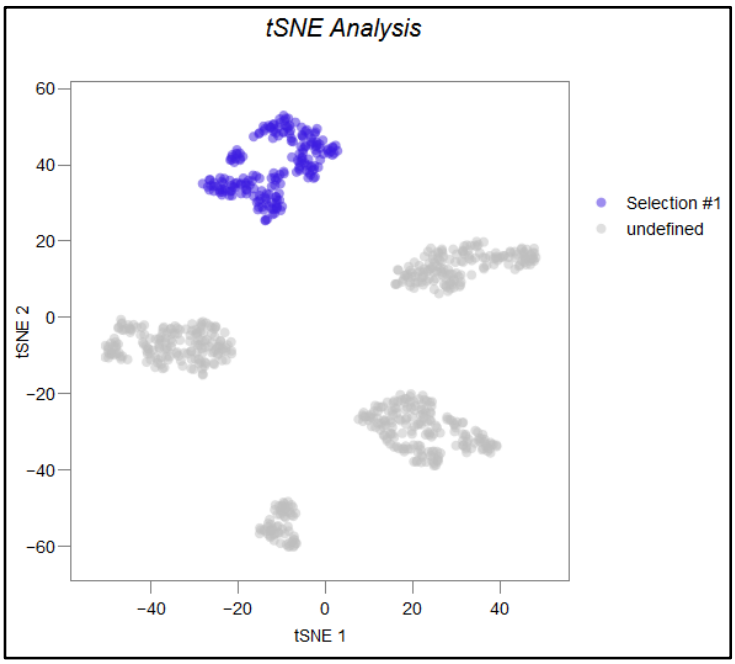


Figure 39. **Illustrating the application of the customizations from iii.1, above.**

vi. Repeat this process, if desired, for other clusters. Or to reset back to the default, click the [Clear custom selections] button.

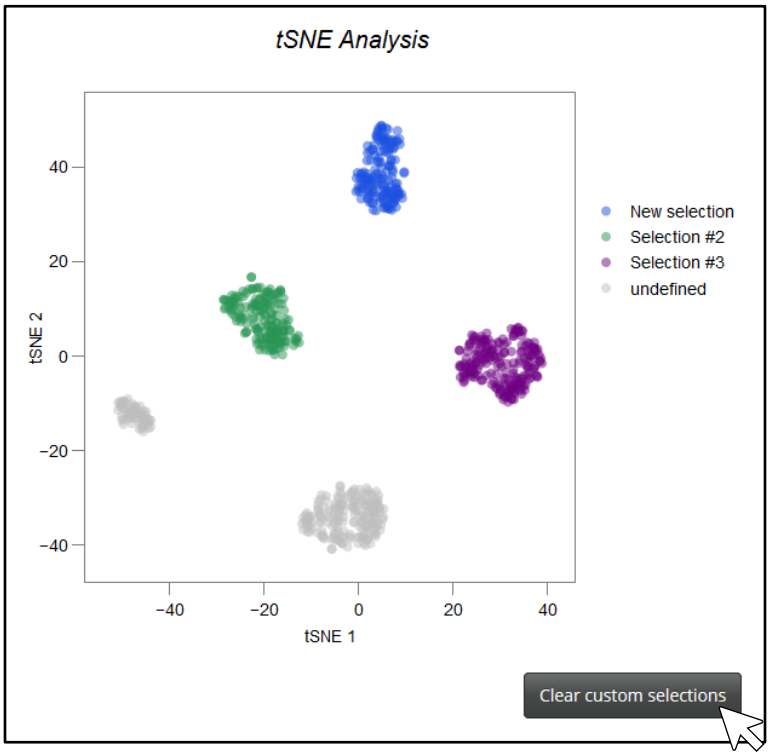


Figure 40. **Multiple custom labeled and tinted clusters and the [Clear custom selections] button.**

F. Export the data

After applying any manipulations to the data from Section VII.E, the data can be saved in its edited form.

1. Expand the **File Management** option in the sidebar menu and click the [Download] button.

Figure 41. The File Management sub-menu.

2. Click either the [Download Plot] or the [Download Data] button to save the information.

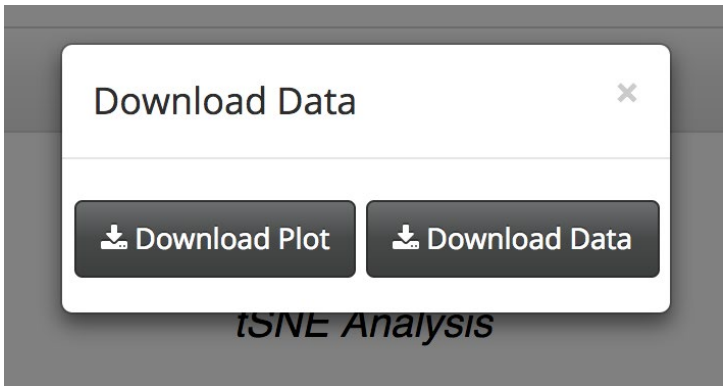


Figure 42. The *Download Data* pop-up menu.

a. **Download Plot** – Open or save the chart image as a `.png` image file

b. **Download Data** – Open or save the processed data as an R-object (`*.rda`) file.

Appendix A. Troubleshooting

If you encounter errors using the hanta reporting tool, please capture a screenshot or the text of the error you may be seeing on the screen and send that plus the relevant log file in to technical_support@takarabio.com.

Table I. Potential issues encountered with the hanta pipeline and the log files related to that area.

Problem area	Log filename
Report generator	`log_hanta`

- If you see this screen, it means that either RStudio can't find the installation of R or R is not installed on the computer. Please see Appendix D or contact your IT specialist for additional assistance.

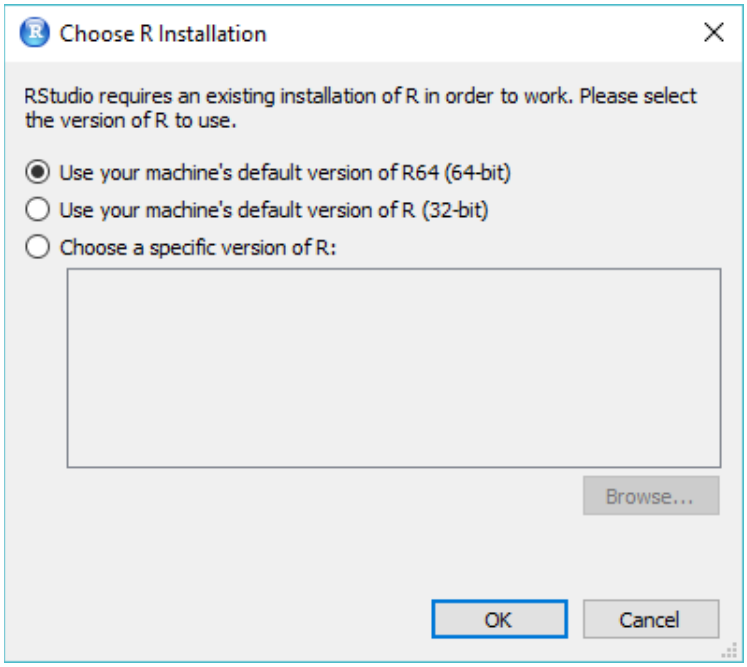


Figure 43. The R installation executable, shown in a Windows Explorer window.

- An error like the following is seen either during installation or an upgrade of hanta software (package name 'glue' is provided as an example only):

```
Error: (converted from warning) cannot remove prior installation of
 package 'glue'
>
```

Figure 44. **Example of an R-dependency package upgrade issue error message.**

the workaround is to manually install the package(s) throwing the error(s).

```
install.packages("<PACKAGENAME>")
```

where `<PACKAGENAME>` is replaced with the name of the package to be re-installed.

Example:

```
install.packages("glue")
```

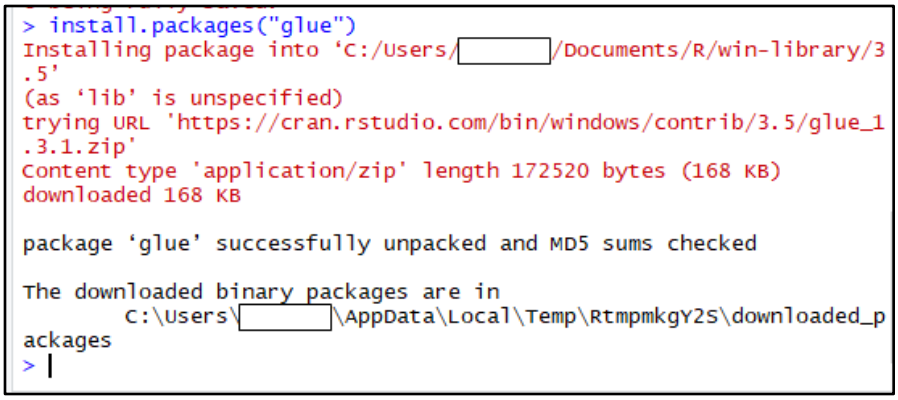


Figure 45. Example of a manual installation of the problem R package as depicted in Figure 44 above. Note the package was successfully unpacked and no error message occurred after the manual install.

Re-run the install/upgrade and repeat as necessary until no additional error messages are seen.

- An error like the following is seen either during installation or an upgrade of hanta software (package name 'processx' is provided as an example only):

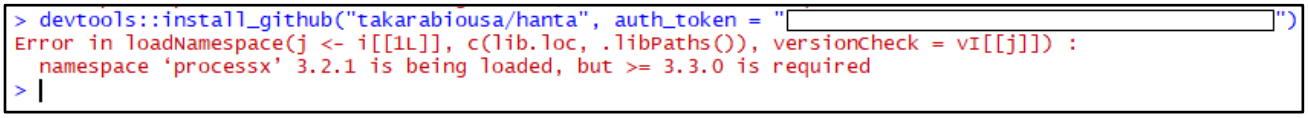


Figure 46. Example of an R-dependency package error during installation.

the workaround is to manually install the package(s) throwing the error(s).

```
install.packages("<PACKAGENAME>")
```

where `<PACKAGENAME>` is replaced with the name of the package to be re-installed.

Example:

```
install.packages("processx")
```

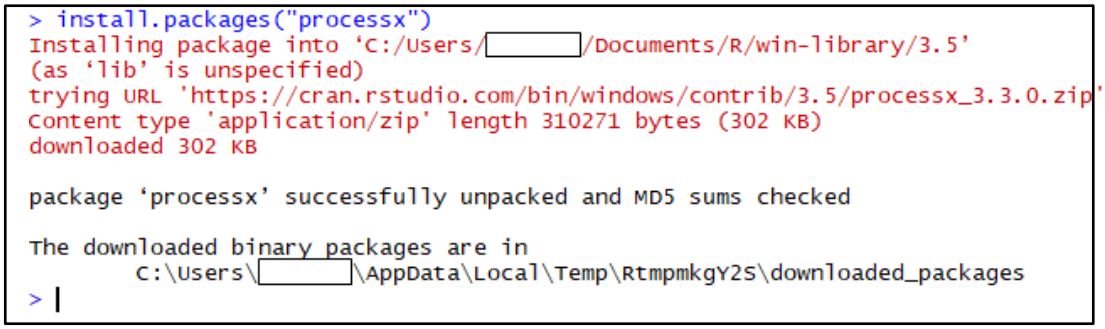


Figure 47. Example of a manual installation of the problem R package as depicted in Figure 46 above. Note the package was successfully unpacked and no error message occurred after the manual install.

Re-run the install/upgrade and repeat as necessary until no additional error messages are seen.

Appendix B. Advanced analysis configuration options

The standard analysis is highly customizable with a wide range of options that can be tailored for experiment-specific needs.

The full syntax, with all options listed with their default values, is:

```
hanta.analysis(gm = NULL, metadata = NULL, mappa_data_loc = NULL, gm_loc = NULL,
  metadata_loc = NULL, gene_info_loc = NULL, cell_type = "all",
  correlation_type = "pearson", parallel_cores = 1, names = NULL, cor_low_thresh = 0,
  cor_high_thresh = 1, quant_or_abs = "quant", gm_norm = TRUE, gm_norm_method = "cpm",
  gm_norm_scale = 10000, gm_log = TRUE, gm_log_base = "ln", qc_cell_abslowcov = 10000,
  qc_cells_abslowgenecount = 300, qc_gene_cellcount = 3, qc_gene_totcov = 100,
  grouping_var = "Sample", pca_filt_method = "top_var", pca_thresh_cut = 0,
  pca_top_genes = 2000, merge_method = "union", correlation_analysis = TRUE,
  cluster_analysis = TRUE, report = TRUE, mappa_data = TRUE, figure_type = "png",
  transpose = FALSE, verbose = FALSE, output_dir = "hanta_output", author = "")
```

The full command syntax above and the list of options in command-required order can also be accessed from within Rstudio with:

```
help("hanta.analysis")
```

The following table is a list of the options, in alphabetical order for easier reference.

Table II. Command-line standard analysis parameter options and definitions.

Option	Description
`author`	Name of individual(s) performing analysis.
`cell_type`	(Optional) Comma separated list of cell types/groups of interest. (i.e., `-t "Positive_Control, K562"`). Defaults to all cell types found in the 'Sample' Column of the metadata/stats file(s).
`cluster_analysis`	TRUE/FALSE, perform clustering analysis (PCA & tSNE). Defaults to `TRUE`.
`cor_high_thresh`	For correlation analysis, filter genes from each cell with expression higher than this value.
`cor_low_thresh`	For correlation analysis, filter genes from each cell with expression less than this value.
`correlation_analysis`	TRUE/FALSE, perform correlation analysis. Defaults to `TRUE`.

correlation_type	Correlation method type. Choose from pearson, spearman, or kendall. Defaults to pearson.
figure_type	Enter pdf or png for figure outputs. Defaults to png.
gene_info_loc	Path to gene_info.csv file from mappa -or- a CSV file with (a) first column containing gene names that match the gene IDs in the gene matrix and (b) another column titled 'Gene_Lengths', with (c) each row containing lengths desired for normalization.
gm	Full path to genematrix.csv file.
gm_loc	(Optional) Useful for passing multiple gene matrices to the function. Plain text file with a list of gene matrices for analysis. Each line of the file must include the full path to the gene matrix (i.e., %GENEMATRIX_PATH%/genematrix.csv).
gm_log	TRUE/FALSE, should gene matrix be log transformed? If TRUE, defaults to natural log 'ln', unless the --gm_log_base option is specified. Defaults to TRUE.
gm_log_base	What log base should be used for log transformation? For natural log select ln. Defaults to ln.
gm_norm	TRUE/FALSE, should gene matrix be normalized? Defaults to TRUE, using counts per million (CPM) unless the gm_norm_method option is also specified.
gm_norm_method	Normalization method. Choose from cpm (counts per million), tpm (transcripts per million) or rpkm (reads per kilobase million).
gm_norm_scale	The scale with which to normalize read counts. Enter median to normalize to the median read coverage across cells. Defaults to 10000 reads for CPM, TPM, RPKM, but can be modified to a different value.
grouping_var	Comma separated list of factors from report files to create groups for analysis. Defaults to Sample, which classifies groups by unique values in the 'Sample' column of the stats/metadata file. Other features including custom user groupings in the stats/metadata file can be entered to facilitate comparisons across different groups. Currently, the function is limited to 20 unique groups.
mappa_data	TRUE/FALSE, is data derived from the mappa pipeline? Specify TRUE if the data files come from mappa, FALSE if not.
mappa_data_loc	Full path to mappa_data folder, obtained from mappa Analysis Pipeline (i.e., %MAPPA_PATH%/mappa_data).

merge_method	How should multiple gene matrices be merged? Choose from intersect or union. Defaults to union, where missing genes are filled in with zero read counts.
metadata	Full path to metadata file (stats.csv file from mappa).
metadata_loc	(Optional) Useful for passing multiple metadata/report files to the function. Plain text file with a list of report files for analysis. Each line of the file must include the full path to the report file for each gene matrix in the -gm_loc option (i.e., %GENEMATRIX_PATH%/genematrix.csv).
names	Optional, comma separated list of names for each input gene matrix (i.e., --names "group1, group2, group3"). If not specified, the basename of each gene matrix file will be used as the name. For example, %MAPPA_OUTPUT%/mappa_output/trial_100_genematrix.csv would be shortened to trial_100
output_dir	Name of the output directory. Defaults to hanta_output in current working directory. The script will not overwrite existing folders, with the exception of the hanta_output folder.
parallel_cores	The number of cores to run in parallel. Not currently supported.
pca_filt_method	Method to choose which genes will be used for clustering analysis. If the option is not specified, all genes are used in the analysis. Choose from: • quant_exp — Filter genes below a quantile value specified in the pca_thresh_cut option • top_var — Select the genes with the highest variance (must also set the pca_top_genes option) • top_exp — Select the highest expressing genes (must also set the pca_top_genes option) Defaults to top_var.
pca_thresh_cut	For clustering analysis, filter genes with expression less than this quantile value (values must be between 0 and 1). Defaults to 0.0 (No genes filtered).
pca_top_genes	Select the number of genes to be used in the clustering analysis. Defaults to 2000.
qc_cell_abslowcov	Define minimum read depth per cell. Defaults to 10000 (10,000 reads). **NOTE**: cells > 3 Median Absolute Deviations (MADs) below the median coverage across cells will automatically be discarded regardless of this value.

`qc_cells_abslowgenecount`	Define minimum gene counts per cell. Defaults to `300` (300 genes). **NOTE:** cells > 3 Median Absolute Deviations (MADs) below the median coverage across cells will automatically be discarded regardless of this value.
`qc_gene_cellcount`	The number of cells a gene must have > 0 expression to be kept in the gene matrix. Defaults to `3`. **NOTE:** cells > 3 Median Absolute Deviations (MADs) below the median coverage across cells will automatically be discarded regardless of this value.
`qc_gene_totcov`	The minimum number of reads for a gene across all cells. Defaults to `100`. **NOTE:** cells > 3 Median Absolute Deviations (MADs) below the median coverage across cells will automatically be discarded regardless of this value.
`quant_or_abs`	Choose `quant` or `abs` to define whether the values for the `cor_low_thresh` and `cor_high_thresh` options are quantile or absolute cutoffs. Defaults to `quant`.
`report`	TRUE/FALSE generate final report. Default is `TRUE`.
`transpose`	TRUE/FALSE. The gene matrix by default should be columns of samples, rows of genes. Toggling this parameter sets the swapped expectation (columns of genes, rows of samples). Defaults to `FALSE`.
`verbose`	TRUE/FALSE, print status to console? Defaults to `TRUE`.

Appendix C. Output file details

Located in `hanta_output/` in the RStudio designated current working directory or the customized `output_dir/`.

Table III. Quality control output files.

Output	Description
`hanta.hanta_output.log`	Log file containing information about the script that was run, time started/finished, total time, the contents of `gm_file_list` and `gm_report_list`, and the options that were selected.
`hanta.raw_data.rda`	R object containing the raw count matrix and metadata associated with the input gene matrix and stats/metadata files.
`hanta.qc_data.rda`	R object containing the quality-controlled gene matrix and metadata objects that have been run through the quality control module. These data have been filtered for poorly performing cells and uninformative genes. The data may also be normalized and log transformed.

Table IV. Correlation analysis output files.

Output	Description
`hanta.correlation_data.rda`	R object containing the correlation matrix, gene matrix, and metadata output from the correlation analysis module, located in the top-level RStudio designated current working directory.
`hanta.cor_stats.csv`	Summary statistics for the intra and intergroup correlation distributions, located in the top-level RStudio designated current working directory.
`hanta.heatmap.png`	Heatmap of the correlation matrix output from the correlation analysis module.
`hanta.boxplot.png`	Boxplot of intragroup correlation distributions.

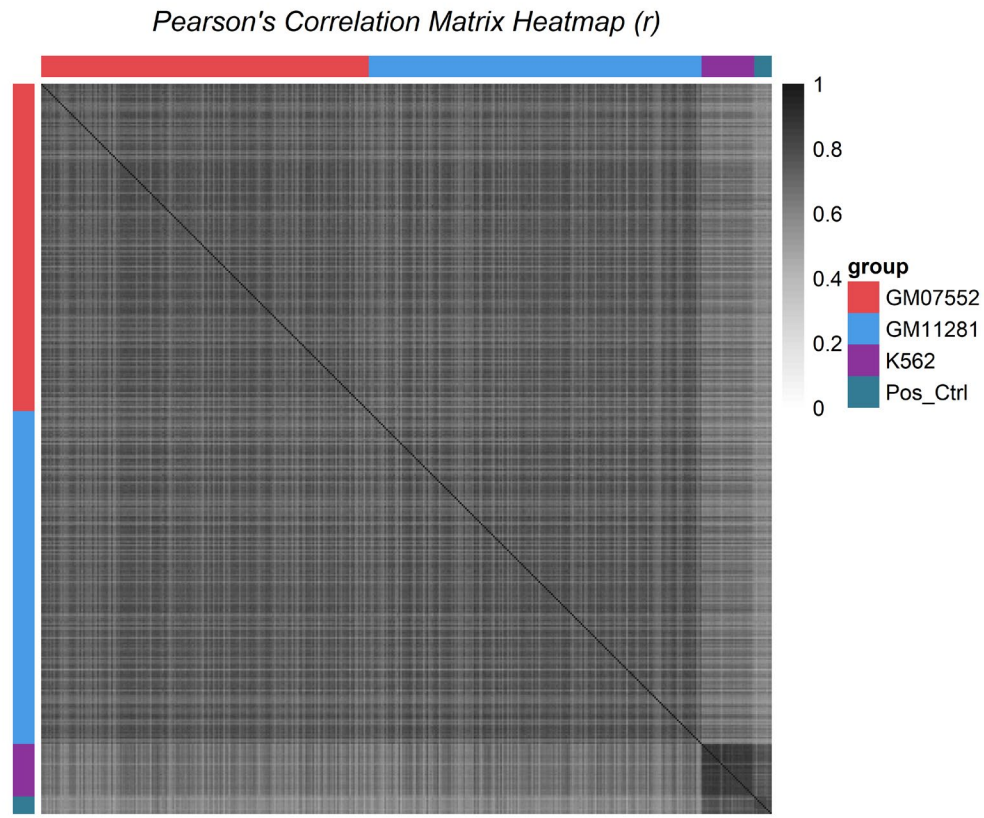


Figure 48. Example `hanta.heatmap.png` output.

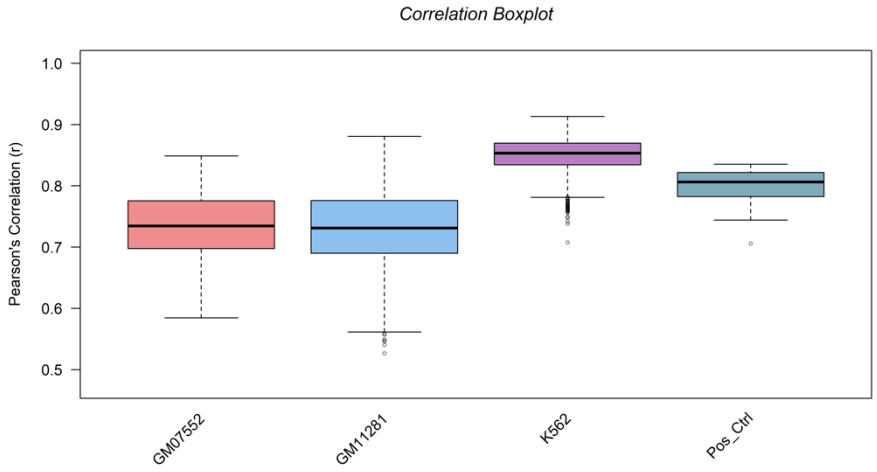


Figure 49. Example `hanta.boxplot.png` output.

Table V. Clustering analysis output files.

Output	Description
`hanta.cluster_data.rda`	R object containing the output of the cluster analysis module, including PCA and tSNE objects
`hanta.pca.png`	Principal Component Analysis Plot
`hanta.tSNE.png`	tSNE Plot

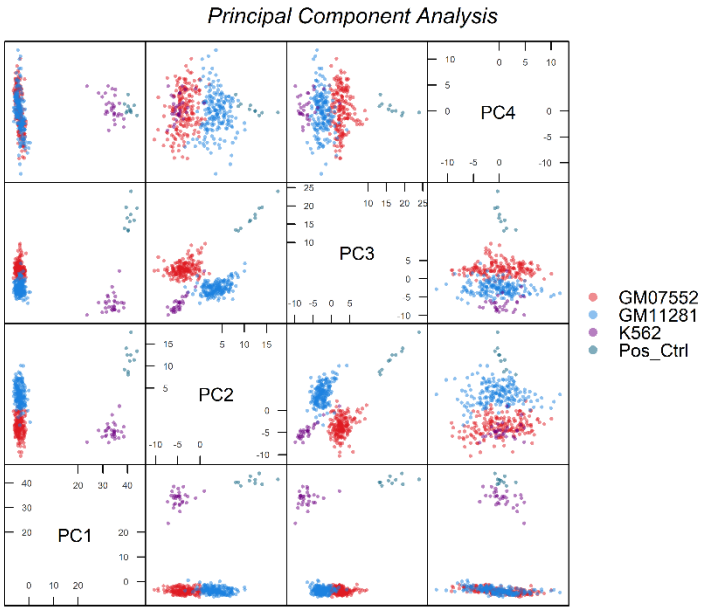


Figure 50. Example `hanta.pca.png` output.

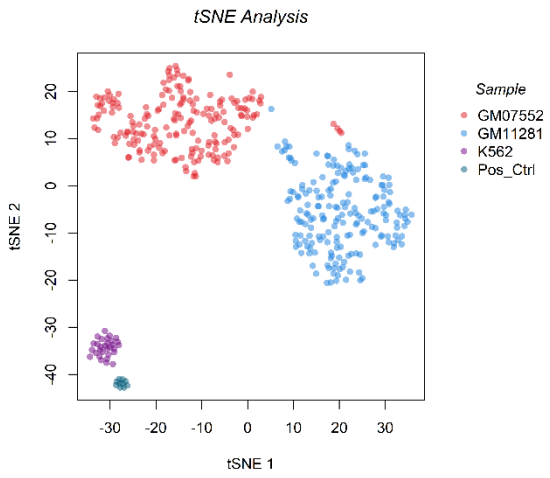


Figure 51. Example `hanta.tSNE.png` output.

Appendix D. Detailed R installation instructions (Windows)

1. Download the R installation executable from cran.cnr.berkeley.edu.

   NOTE: Please use version 3.5.0 or higher.

2. Run the installation executable.

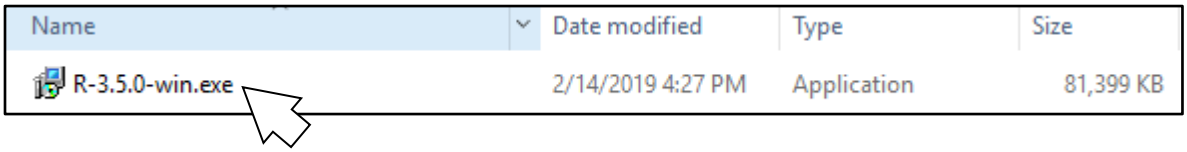


Figure 52. The R installation executable, shown in a Windows Explorer window.

3. If prompted with this option, click [Run].

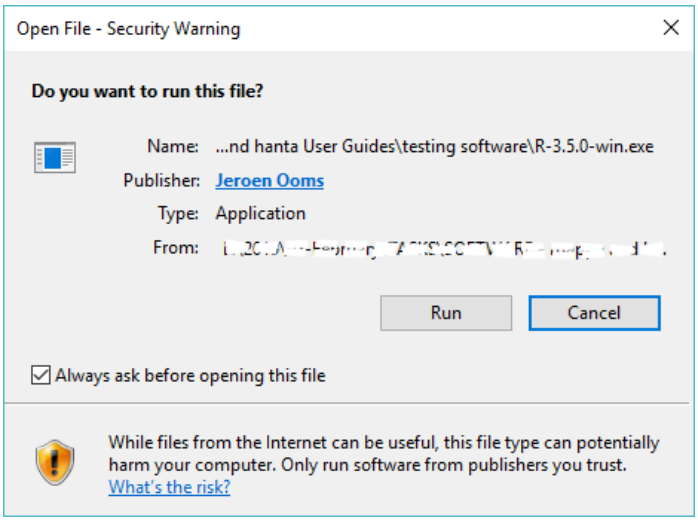


Figure 53. Standard Windows installation security warning pop-up window.

4. Select the language to use during the install process.

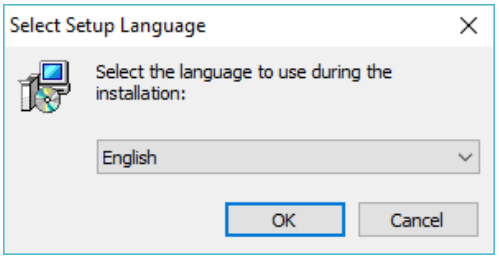


Figure 54. Setup language prompt.

5. Read through the license agreement and click [Next] to accept it.

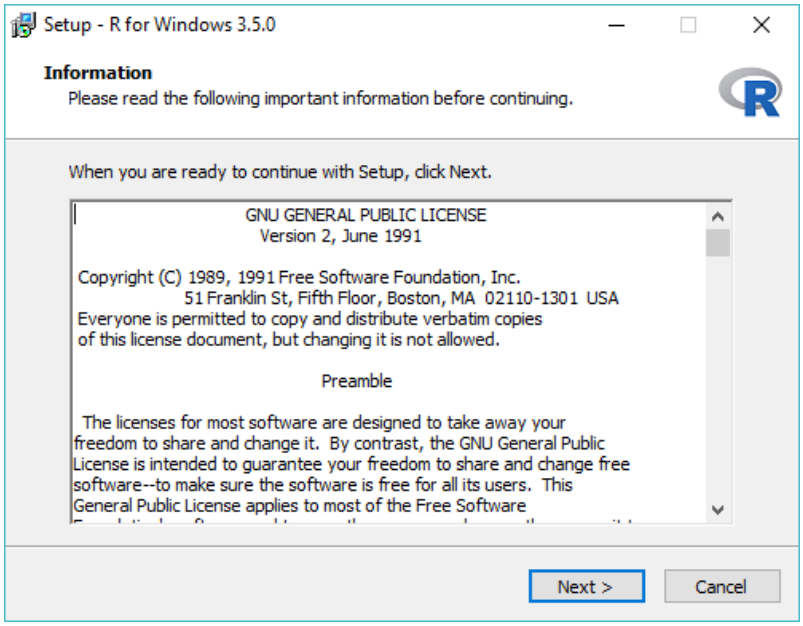


Figure 55. Example R license agreement prompt.

6. Select the directory location to install into.

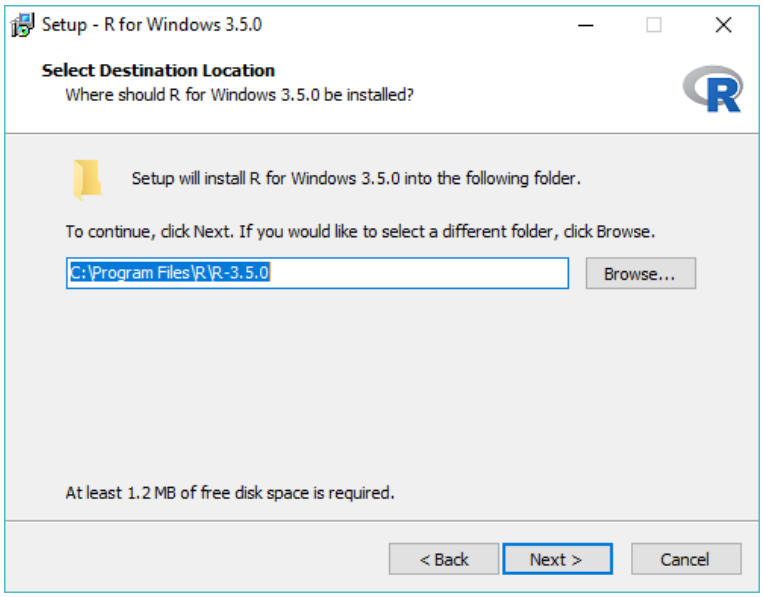


Figure 56. Default *Select Destination Location* window during the R install.

The default location can be accepted or click [Browse] to choose a different install location.

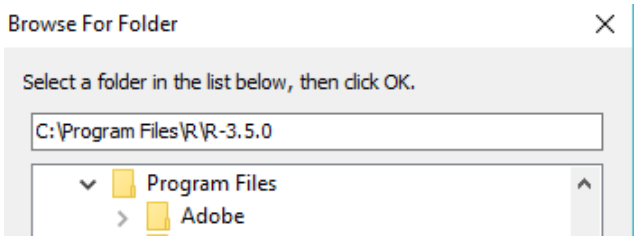


Figure 57. Example view if the [Browse] button is selected from Figure 56.

7. At the *Select Components* window, keep all boxes checked (default) and click [Next].

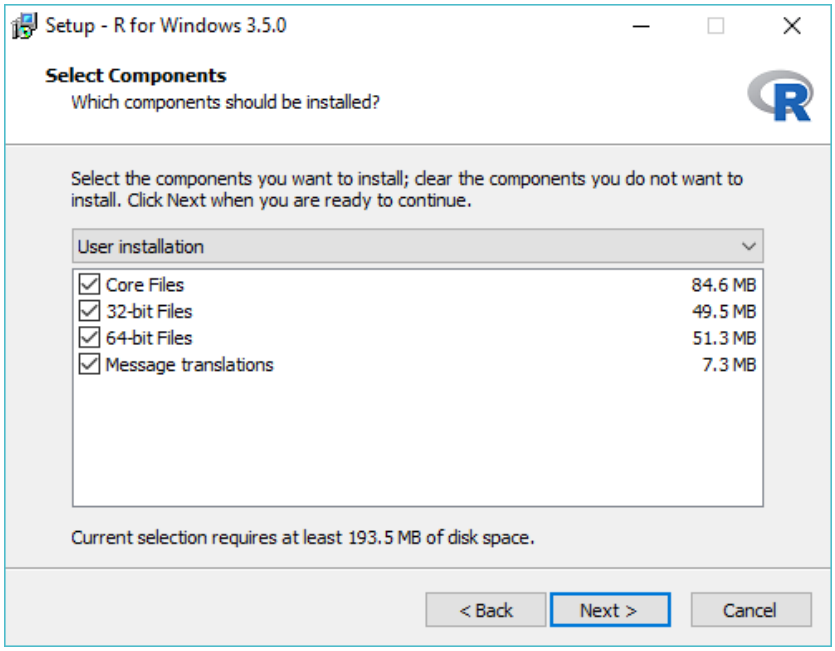


Figure 58. ***Select Components*** **window during the R install process.**

8. At the *Startup options* window, accept the defaults (the "No" radio button selected) and click [Next].

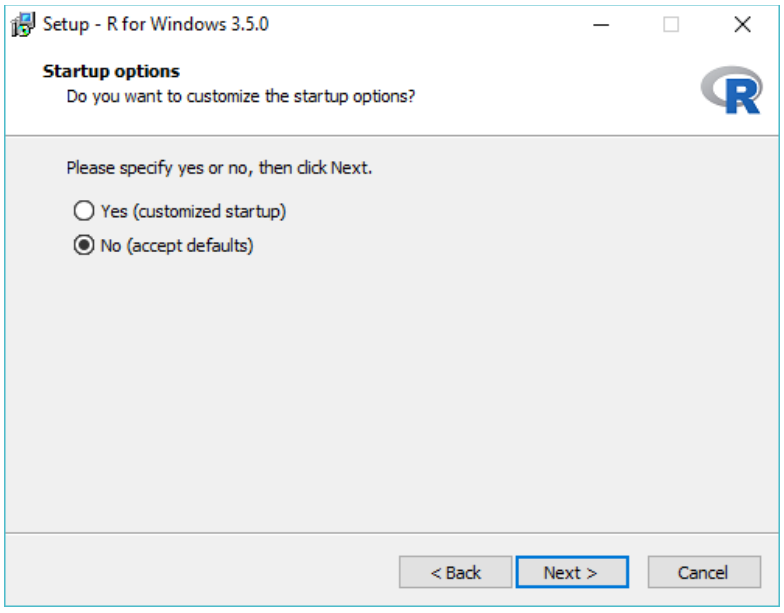


Figure 59. ***Startup options*** **window during the R install process.**

9. Select the Start Menu folder for R.

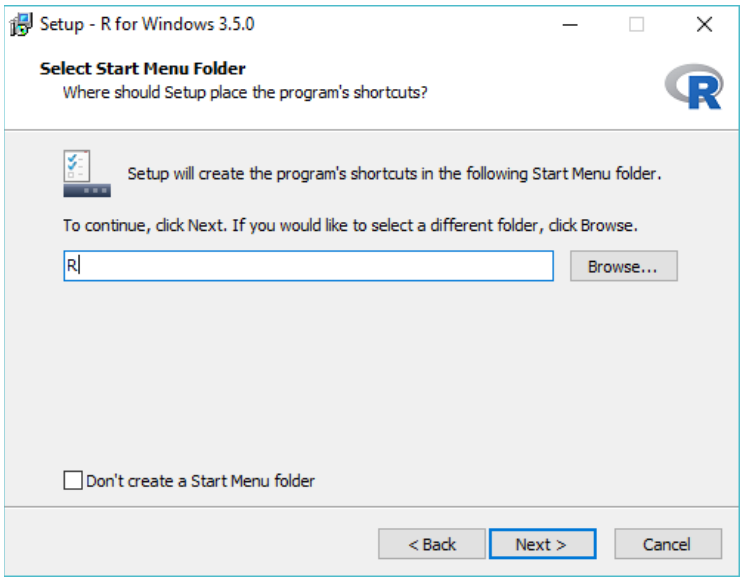


Figure 60. ***Select Start Menu Folder*** **window during the R install process.**

Or to not create a folder, check the box.

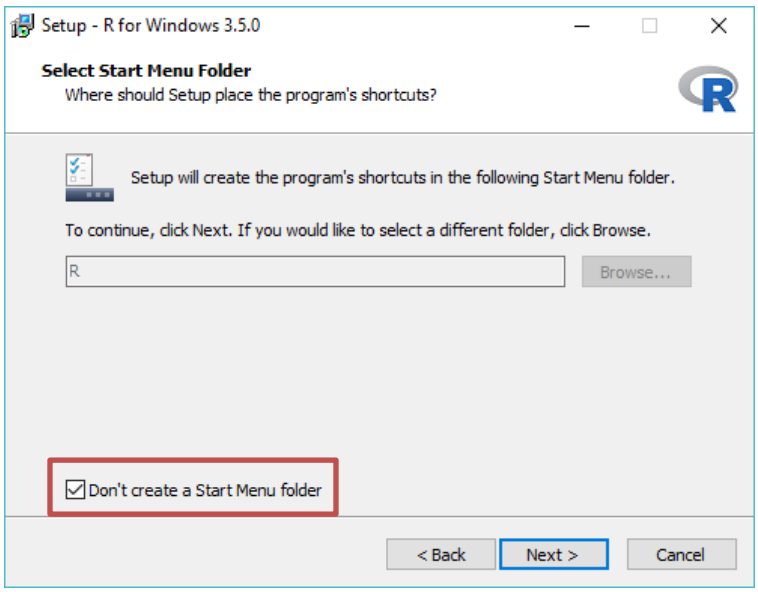


Figure 61. Check box to not create a Start Menu folder for R during the install.

10. Select any additional setup options you'd like to do by checking or unchecking the box next to it Click [Back] to change any parameters or [Next] to begin the install.

> **NOTE**: We recommend using the default options, i.e., the boxes pre-checked as part of the installation process.

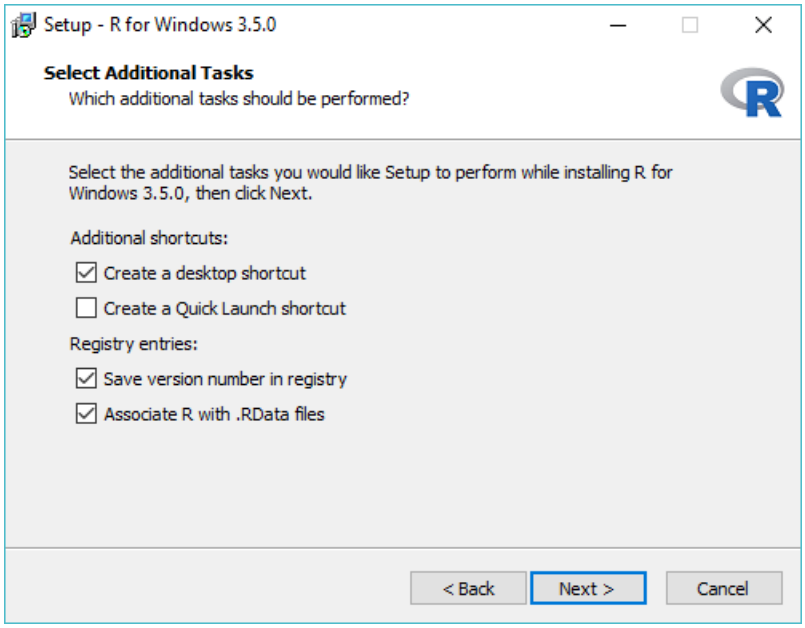


Figure 62. ***Select Additional Tasks*** **window during the R install process.**

11. R will begin to install.

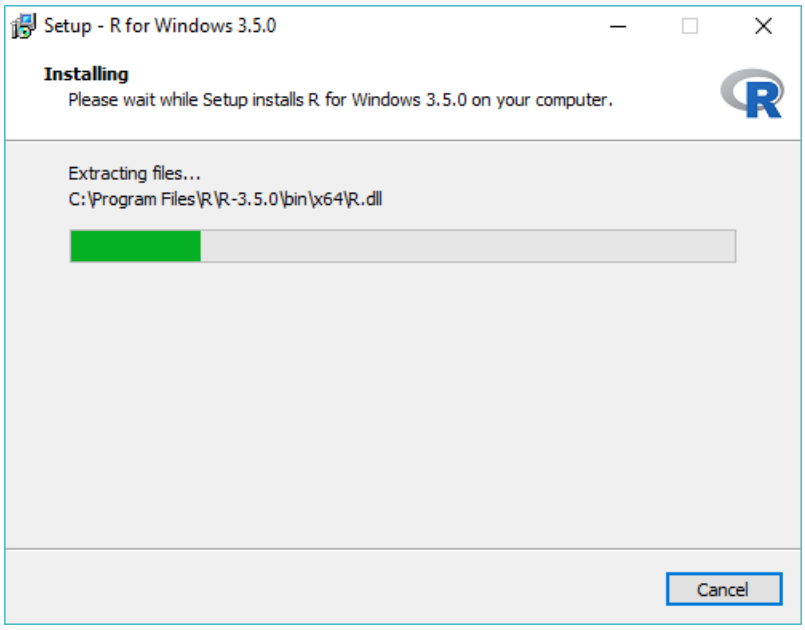


Figure 63. Installation progress window example.

12. Once complete, the following screen will appear. Click [Finish] to complete the installation.

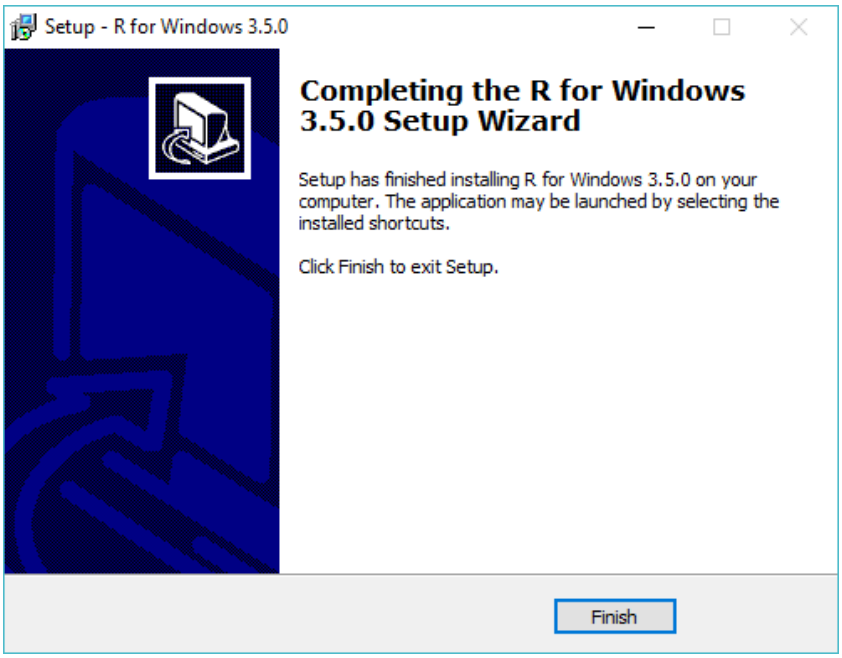


Figure 64. Setup completed window for the R installation.

Appendix E. Detailed Rtools installation instructions

REMINDER: Rtools is a Windows-only prerequisite.

1. Download the free `Rtools` install package from cran.cnr.berkeley.edu/bin/windows/Rtools.
2. Run the installation executable.

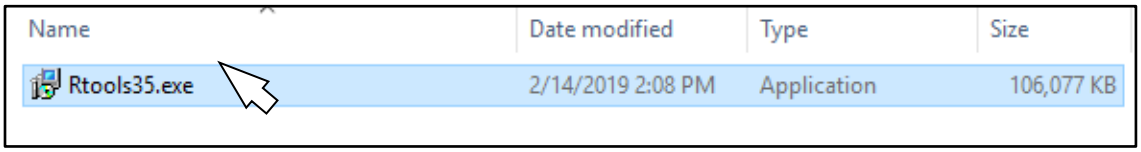


Figure 65. The Rtools installation executable, shown in a Windows Explorer window.

3. If prompted with this option, click [Run].

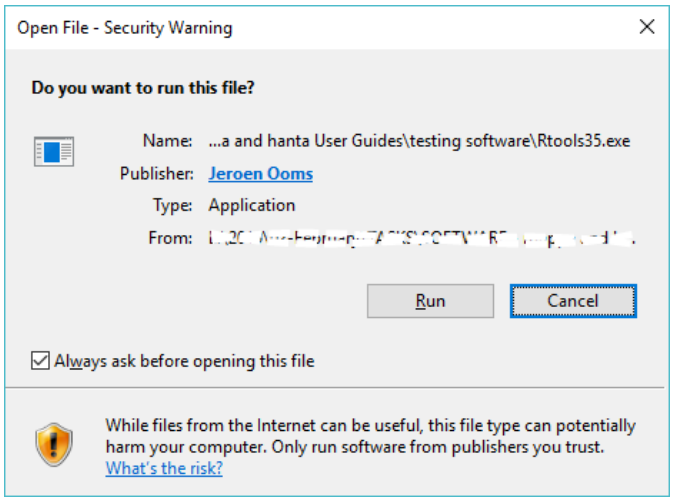


Figure 66. Standard Windows installation *Security Warning* pop-up window.

4. Select the language to use during the install process.

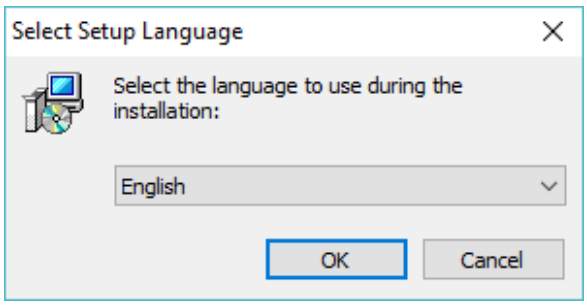


Figure 67. *Select Setup Language* prompt.

5. Read through the license agreement and click [Next] to accept it.

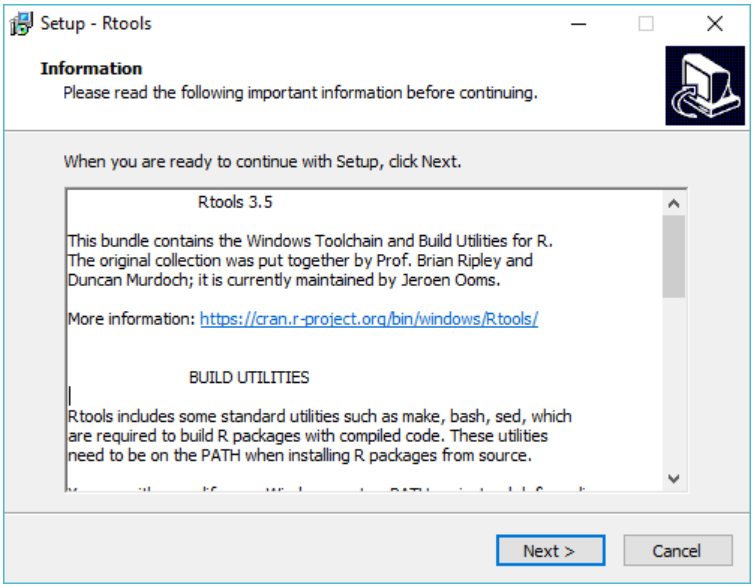


Figure 68. Example Rtools license agreement prompt.

6. Select the directory location to install into.

> **NOTE**: On Windows, Rtools must be installed in a file path with directory names which do not include spaces. (i.e., it cannot be installed in `C:\Program Files\`, but could be installed in `C:\Program\`). Installing it in a file path with spaces in the directory names will cause the hanta software installation to fail.
>
> If Rtools is installed in such a location on the target computer, please uninstall Rtools and re-install in a folder with a path that conforms to these requirements.

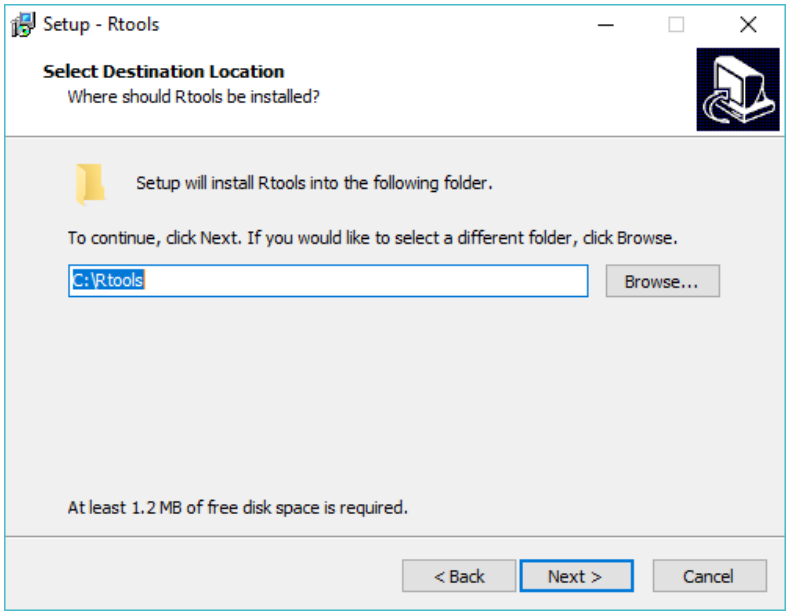


Figure 69. Default *Select Destination Location* window during the Rtools install.

The default location can be accepted or click [Browse] to choose a different install location.

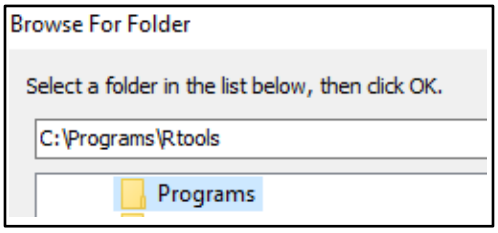


Figure 70. Example view if the [Browse] button is selected from Figure 69.

7. At the *Select Components* window, keep all boxes checked (default). Please note the free disk space requirements and ensure there is enough hard drive space for the drive letter Rtools will be installed to. Once verified, click [Next].

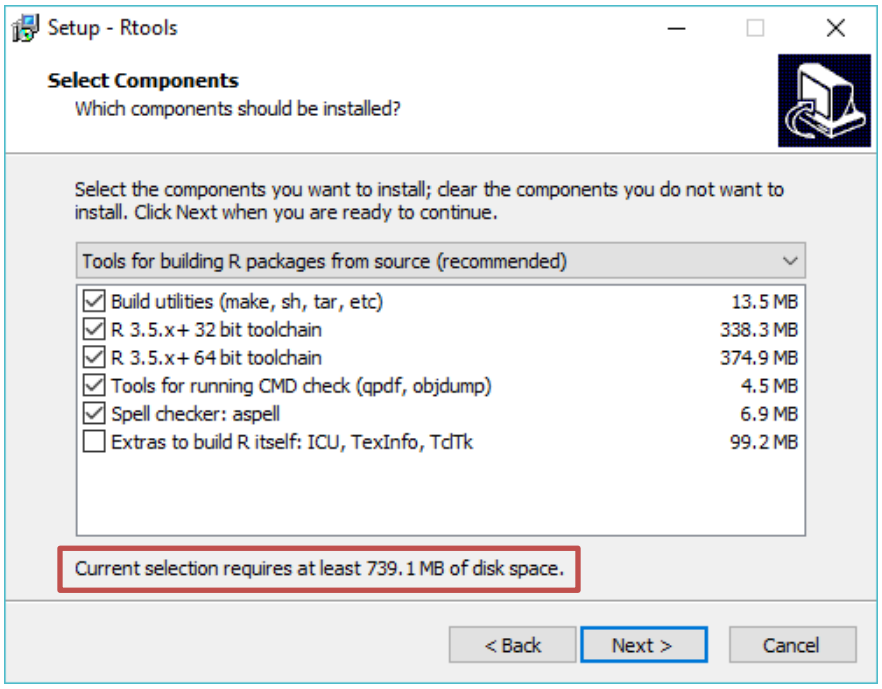


Figure 71. ***Select Components*** **window during the Rtools install process.**

8. At the *Select Additional Tasks* step, ensure "Add rtools to system PATH" option is checked. Once checked, hit [Next].

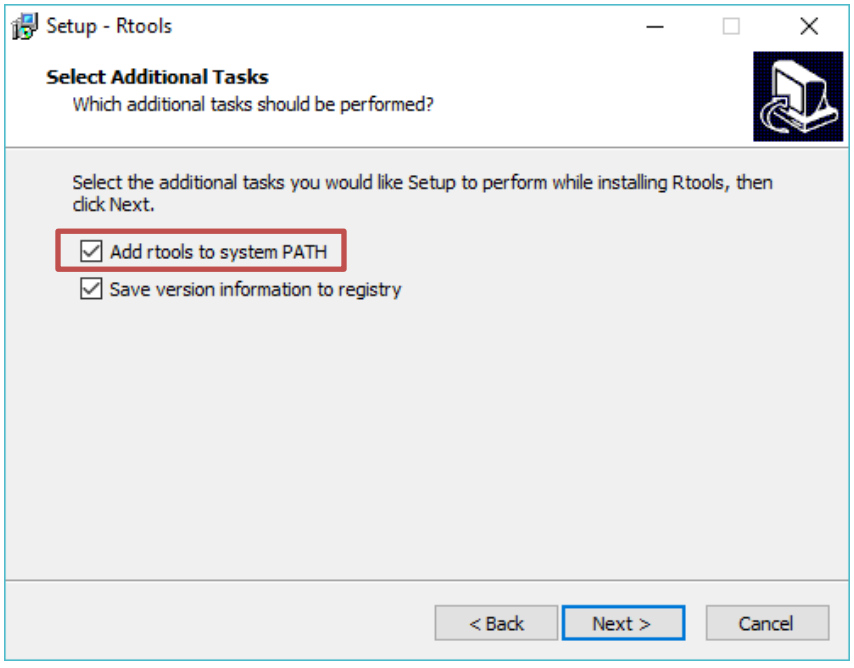


Figure 72. ***Select Additional Tasks*** **window during the Rtools install process. "Add rtools to system PATH" must be selected as a prerequisite for hanta software.**

9. If you need to edit the path, do so on the next screen; by default, you shouldn't have to. Once the `Rtools\bin` directory path is confirmed, click [Next].

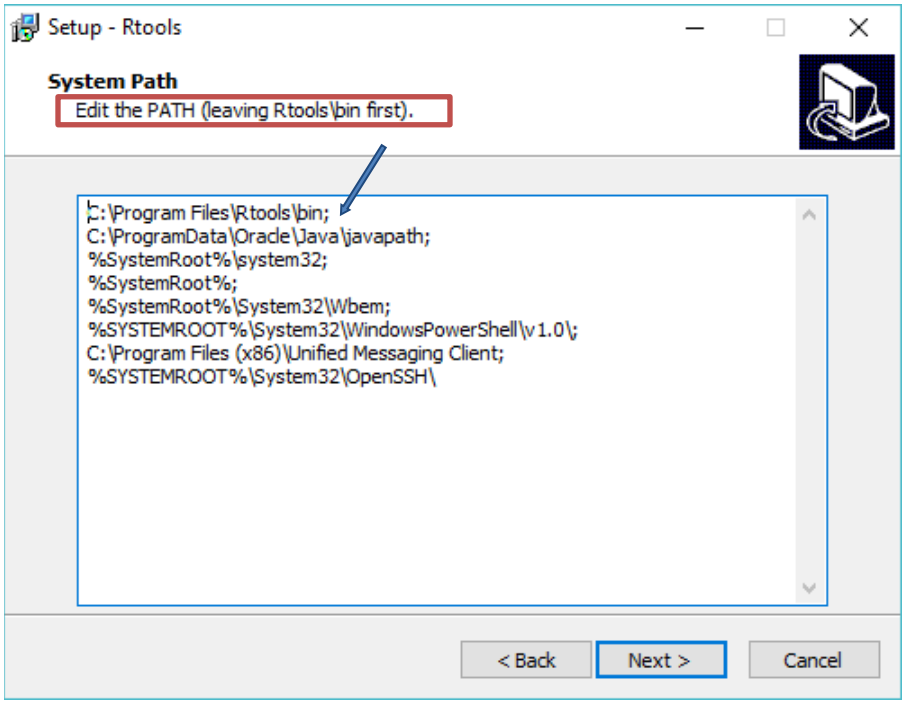


Figure 73. The *System Path* window of the Rtools install process. Ensure the path to `Rtools\bin` is correct and the first row of the list.

10. The next window is *Ready to Install*. Click [Back] to change any parameters or [Install] to proceed.

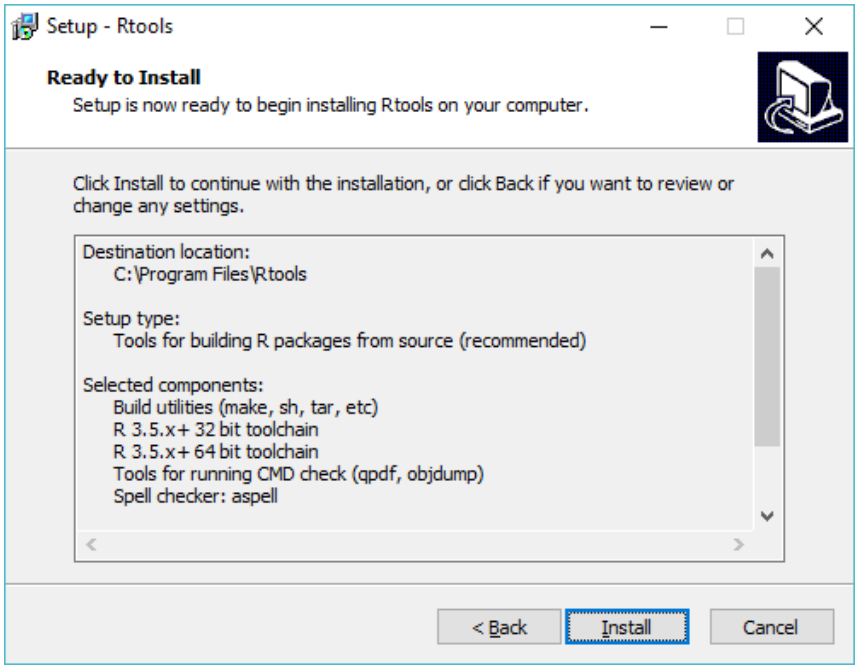


Figure 74. *Ready to Install* window of the Rtools install process.

11. Rtools will begin to install.

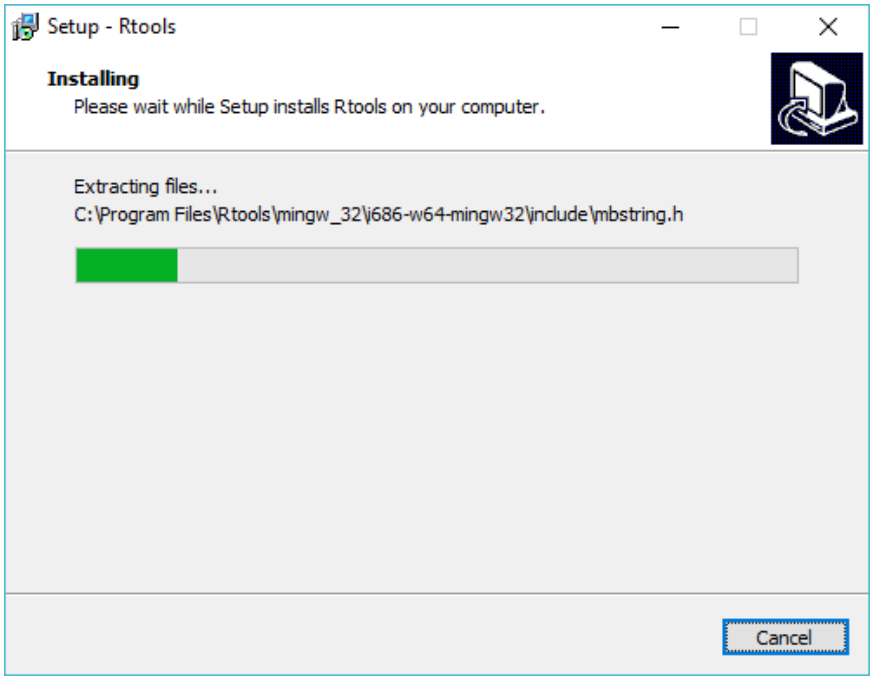


Figure 75. Installation progress window example.

12. Once complete, the following screen will appear. Click [Finish] to complete the installation.

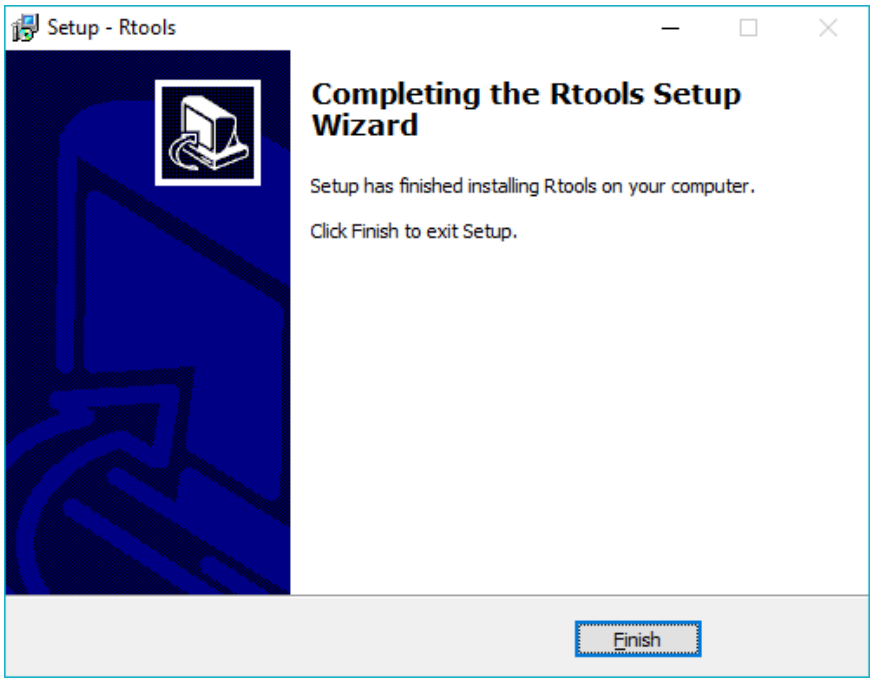


Figure 76. *Setup complete* window for the Rtools installation.

Appendix F. Detailed RStudio installation and configuration instructions

A. Installation

1. Download the free "RStudio Desktop – Open Source License" version of the install package from rstudio.com.

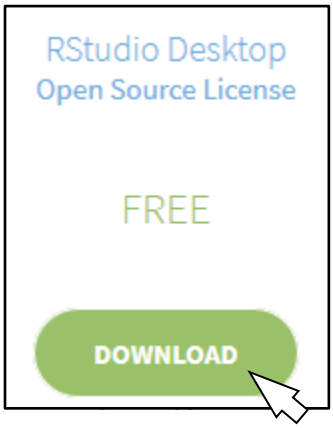


Figure 77. Screenshot showing the selection to make at the RStudio download page.

2. Run the installation executable.

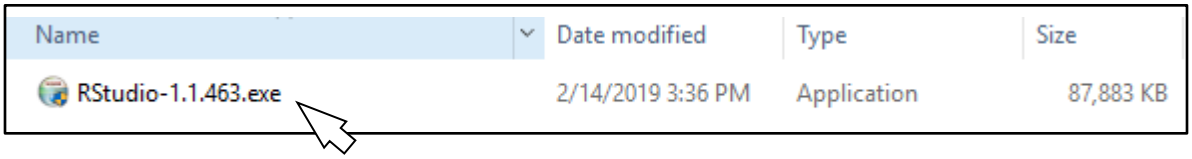


Figure 78. The RStudio installation executable, shown in a Windows Explorer window.

3. If prompted with this option, click [Run].

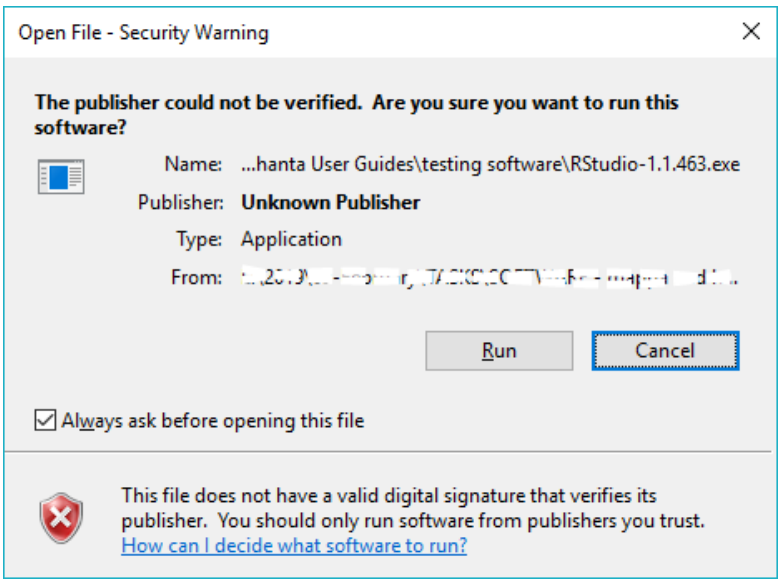


Figure 79. Standard Windows installation *Security Warning* pop-up window.

4. When prompted with this window, click [Next] to proceed with the setup.

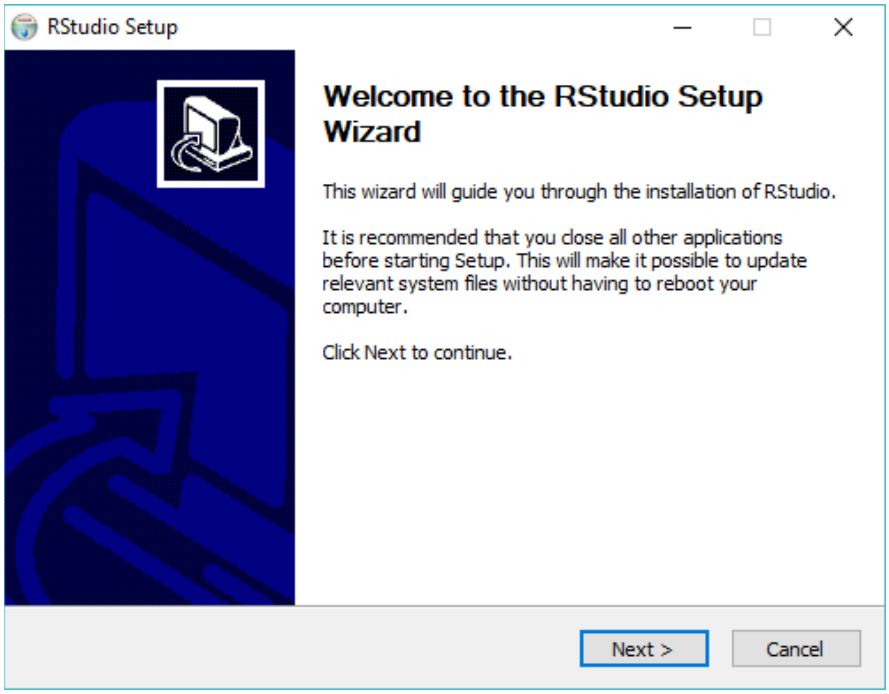


Figure 80. ***RStudio Setup Wizard*** **window.**

5. Select the directory location to install into. Please note the free disk space requirements.

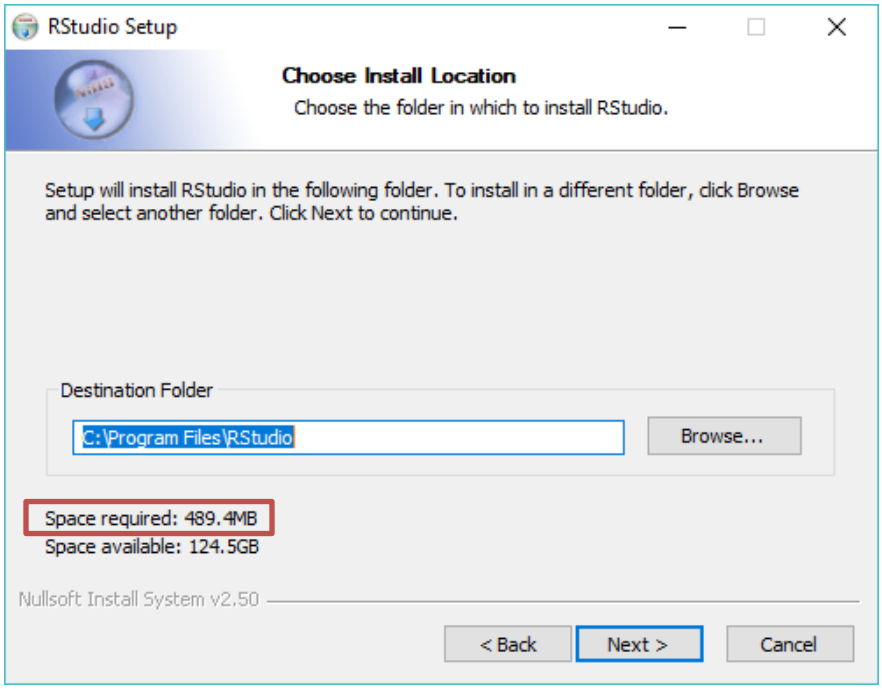


Figure 81. Default ***Choose Install Location*** **window during the RStudio install. Double-check the free disk space requirement and compare against the space available on the computer it is being installed on.**

The default location can be accepted or click [Browse] to choose a different install location.

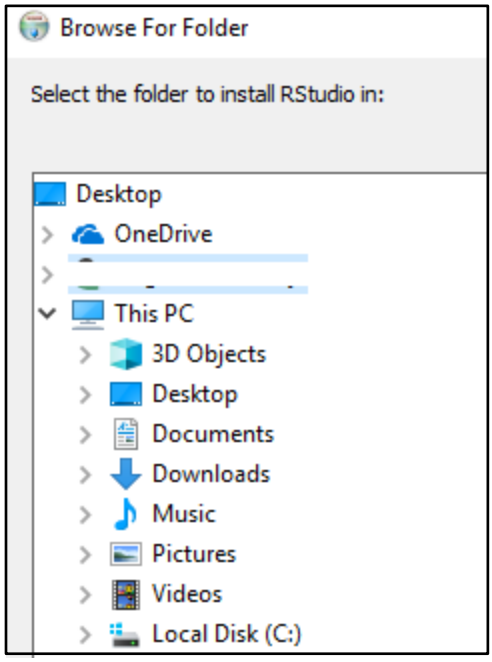


Figure 82. Example view if the [Browse] button is selected from Figure 81.

Once the folder to install in is selected, click [Next].

6. You'll next be prompted to specify which `Start Menu` folder to place RStudio. Either accept the default or select one from the list.

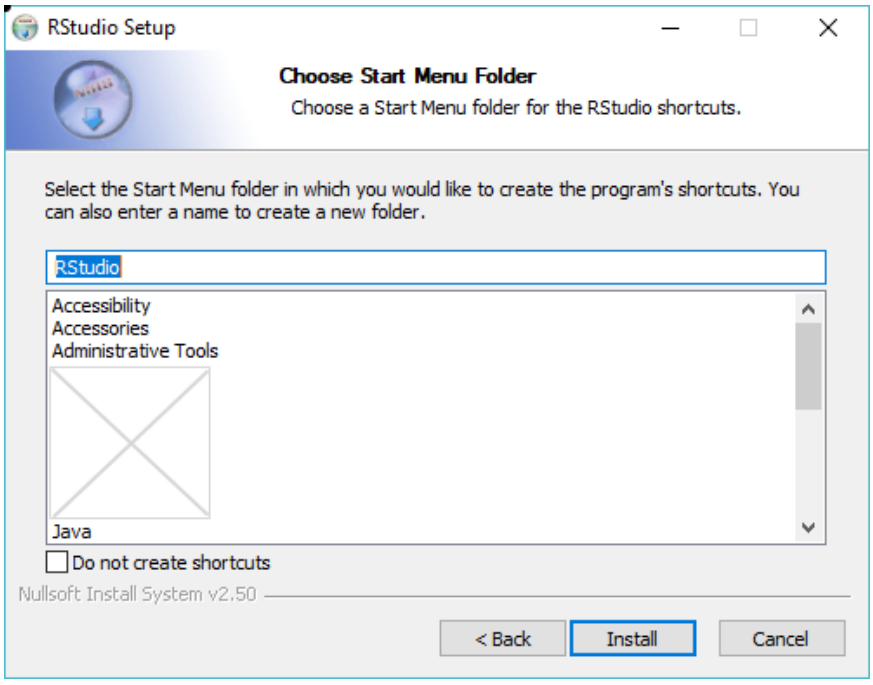


Figure 83. Choose *Start Menu Folder* window during RStudio install.

If you don't want a shortcut created, click the box next to "Do not create shortcuts".

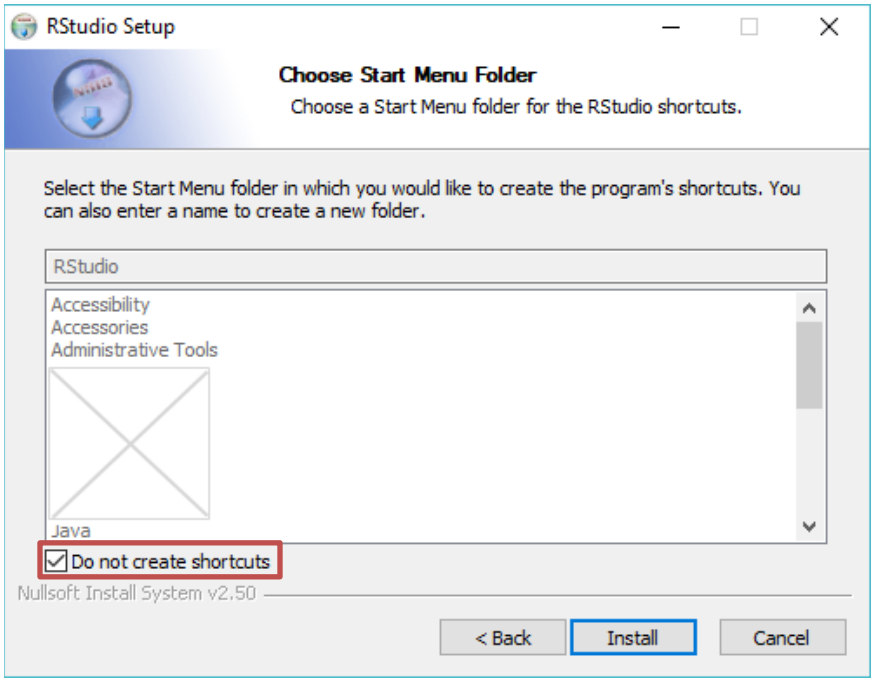


Figure 84. ***Choose Start Menu Folder*** **window, checking the "Do not create shortcuts" box. The option to specify a `Start Menu` folder also grays out and cannot be edited when checked.**

After making the selection, click [Install] to start the installation process.

7. RStudio will begin to install.

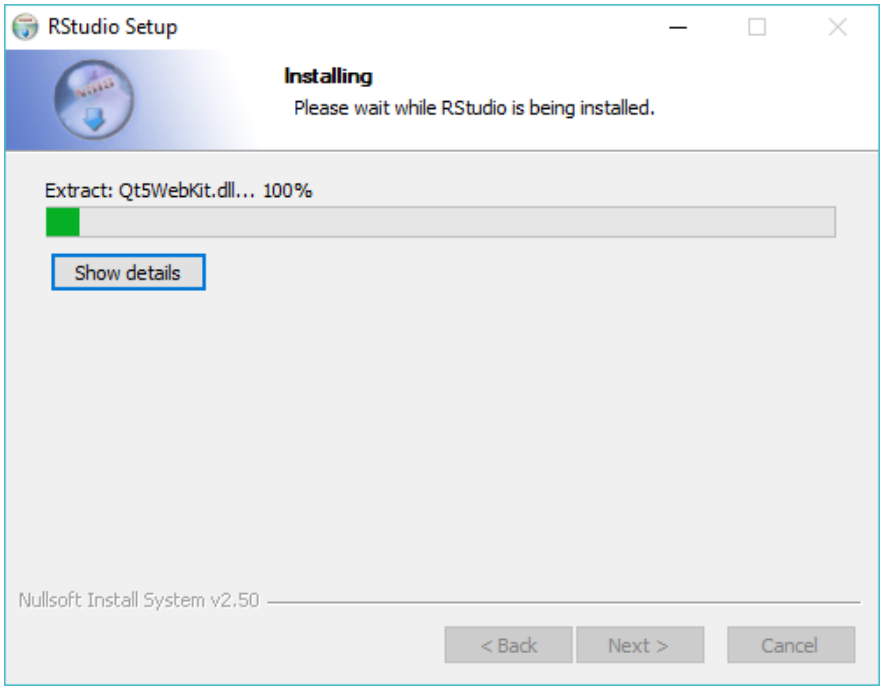


Figure 85. Installation progress example.

8. Once complete, the following screen will appear. Click [Finish] to complete the installation.

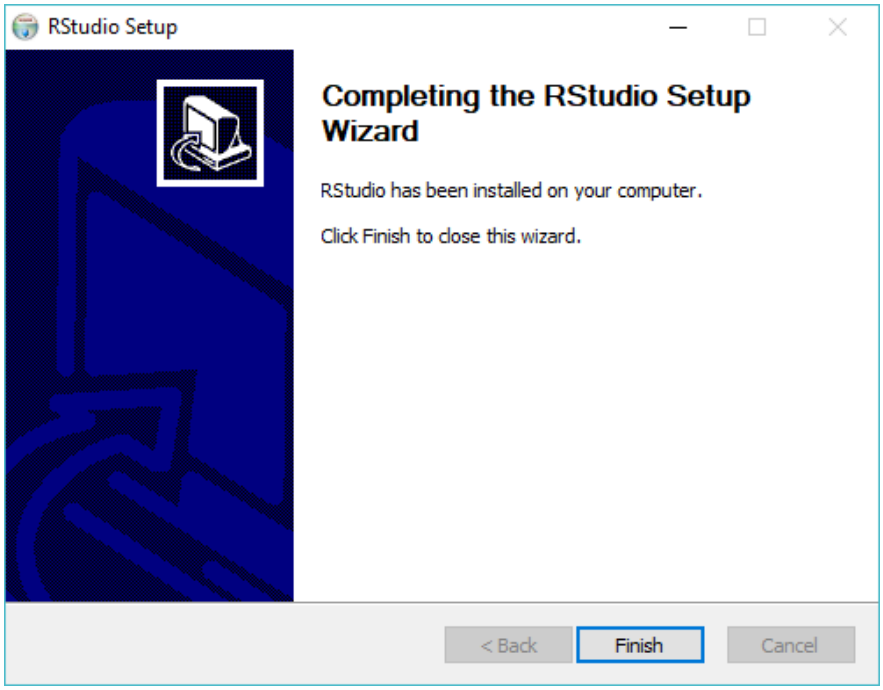


Figure 86. ***Setup complete*** **window for the RStudio installation.**

B. Configuration

1. Find RStudio on the computer it was installed on and run it.

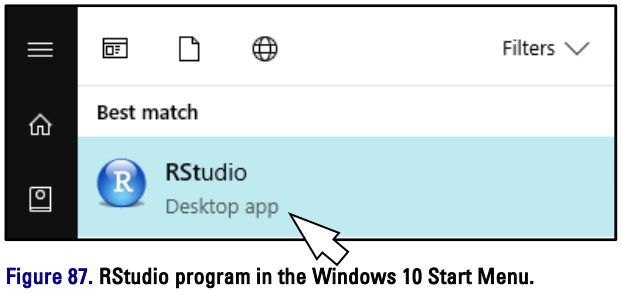


Figure 87. RStudio program in the Windows 10 Start Menu.

2. When RStudio runs, you will see a user interface like this:

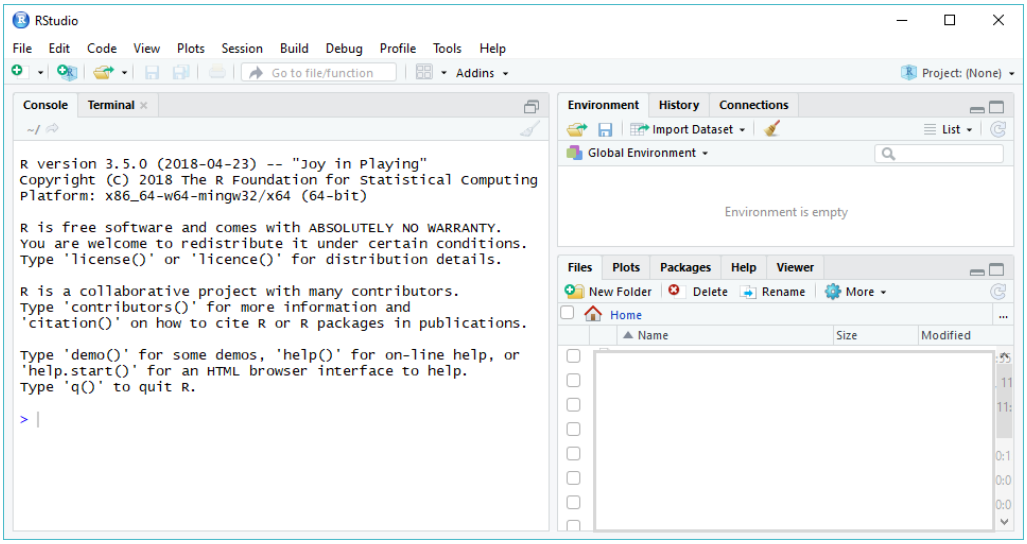


Figure 88. RStudio graphical user interface (GUI).

3. The default Working Directory (bottom right hand window) will be `This PC > Documents`:

Figure 89. The default RStudio Working Directory is `This PC > Documents`, shown here in Windows Explorer.

To select a new Working directory:

a. Through Windows Explorer, identify the new folder you would like to make the Working Directory; if it does not exist, create it.

b. In RStudio, click on the ellipsis icon for **Go to directory**:

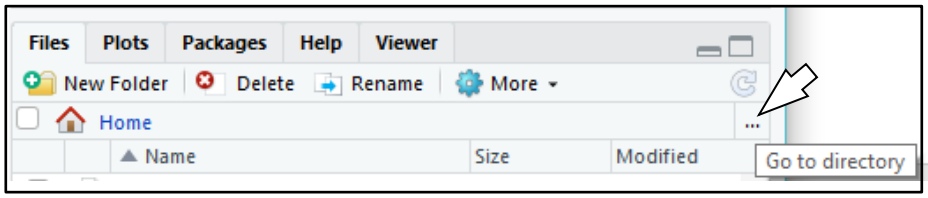


Figure 90. Location of the ellipsis icon for Go to directory in the RStudio GUI.

c. Browse to and select the desired folder:

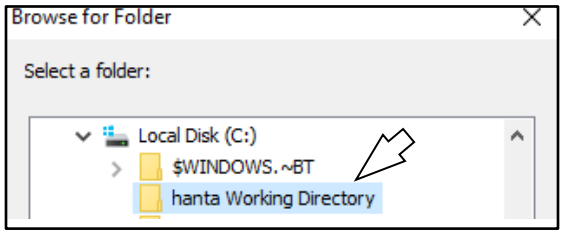


Figure 91. Selecting a new folder to customize the RStudio Working Directory.

d. Click the [OK] button.

The selected folder will now display in the lower right-hand window

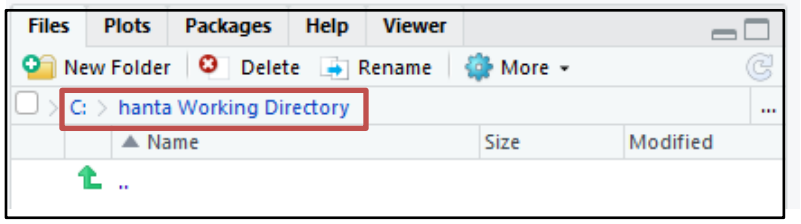


Figure 92. **The customized Working Directory displays in the folder window in the RStudio GUI.**

e. To set this as the new default, click on the **More** menu icon and select **Set as Working Directory** from the drop-down menu.

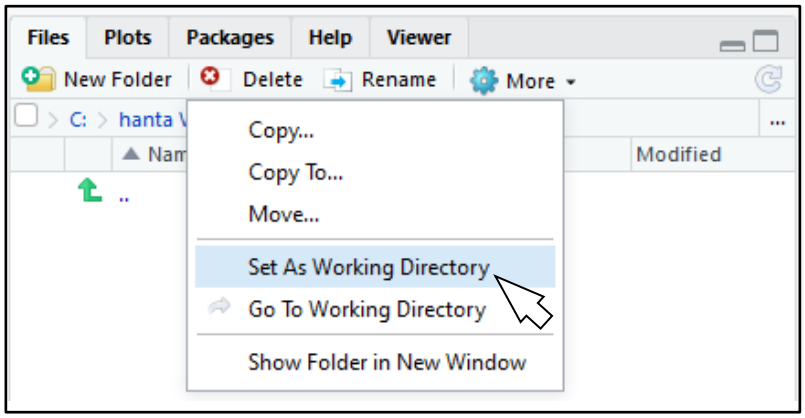


Figure 93. **The drop-down menu under the More menu to set a new folder as the default RStudio Working Directory.**

f. You will see a message in the Console window (left-hand side) similar to this:

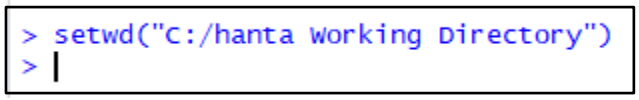


Figure 94. **Verification in the Console window that the new working directory has been set.**

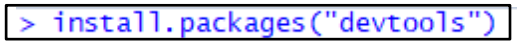
Appendix G. Detailed devtools installation instructions

Devtools may be installed in RStudio either via command-line in the Console window or via a menu-prompted GUI process.

A. RStudio Console

1. Run the RStudio GUI if it is not already running (see Appendix F.B, steps 1 and 2 for more information).
2. Type the following in to the RStudio Console prompt:

```
install.packages("devtools")
```



Text similar to the following will write to the screen:

```
> install.packages("devtools")
Installing package into 'C:/Users/[redacted]/Documents/R/win-library/
3.5'
(as 'lib' is unspecified)
also installing the dependencies 'sys', 'ps', 'askpass', 'magrittr
', 'backports', 'Rcpp', 'ini', 'processx', 'R6', 'assertthat', 'cr
ayon', 'curl', 'mime', 'openssl', 'desc', 'prettyunits', 'rprojroo
t', 'rlang', 'xopen', 'clipr', 'clisymbols', 'fs', 'gh', 'glue', '
whisker', 'callr', 'cli', 'digest', 'git2r', 'httr', 'jsonlite', '
memoise', 'pkgbuild', 'pkgload', 'rcmdcheck', 'remotes', 'rstudioa
pi', 'sessioninfo', 'usethis', 'withr'


  There are binary versions available but the source
  versions are later:
          binary source needs_compilation
R6         2.3.0  2.4.0             FALSE
rcmdcheck  1.3.1  1.3.2             FALSE

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/sys_2
.1.zip'
```

Figure 95. The `devtools` installation process commencing, seen in the RStudio Console window.

3. Let it continue to run until it returns to a > prompt, as in the example in Figure 96 (below).

```
** building package indices
** testing if installed package can be loaded
*** arch - i386
*** arch - x64
* DONE (rcmdcheck)
In R CMD INSTALL

The downloaded source packages are in
        'C:\Users\[redacted]\AppData\Local\Temp\RtmpYZySkE\downloaded
_packages'
>
```

Figure 96. An example of the Console window at the end of a successful installation of the devtool package.

B. Packages menu

Devtools may also be installed via a menu-driven UI system, if preferred.

1. In RStudio, click the *Packages* tab on the lower right-hand side of the GUI.

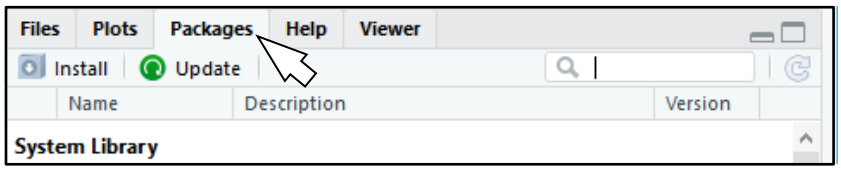


Figure 97. The location of the *Packages* tab in the RStudio GUI.

2. Under Packages, select **Install**.

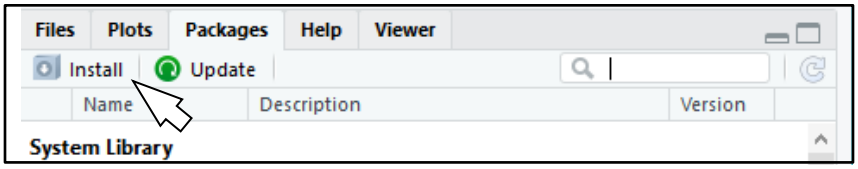


Figure 98. The Install icon location.

The *Install Packages* window pops up:

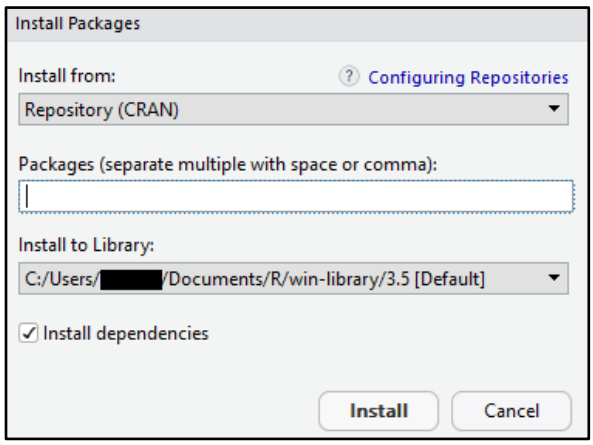


Figure 99. The *Install Packages* pop-up menu.

3. Type the string `dev` into the "Packages" box and autocomplete options will appear in a drop-down menu under the box. Select 'devtools'.

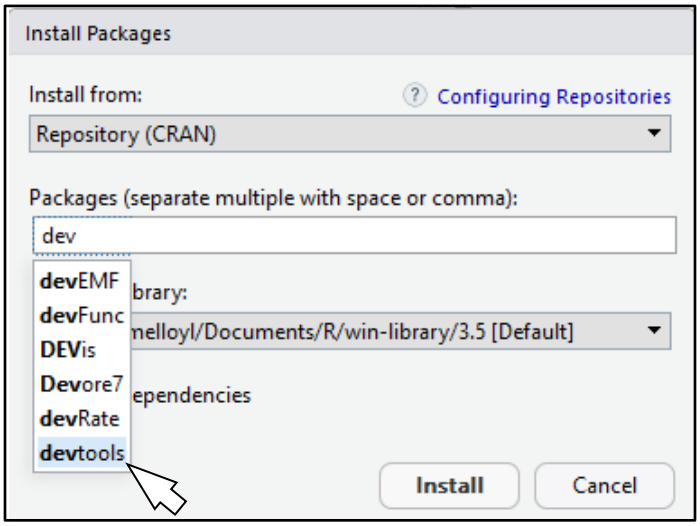


Figure 100. The autocomplete option drop-down with the devtools selection highlighted.

4. Click the [Install] button.

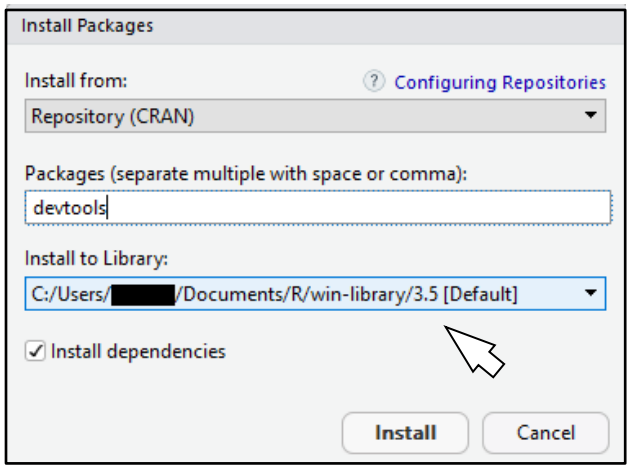


Figure 101. The *Install Packages* window with devtools selected and ready to Install.

5. In the Console window of RStudio, text similar to Figures 95 and 96 (above) will write to the screen.
6. After the installation ends, “devtools” will display in the list of installed tools under the Packages tab.

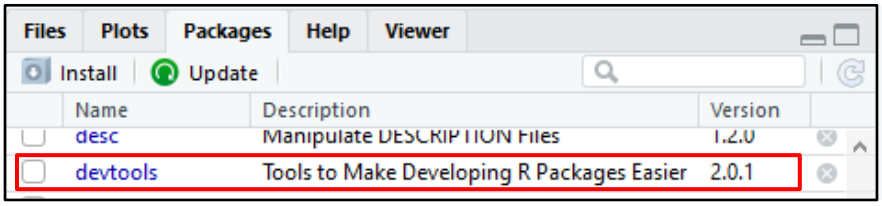


Figure 102. devtools displaying in the list of installed packages in RStudio.

Contact Us	
Customer Service/Ordering	**Technical Support**
tel: 800.662.2566 (toll-free)	tel: 800.662.2566 (toll-free)
fax: 800.424.1350 (toll-free)	fax: 800.424.1350 (toll-free)
web: takarabio.com	web: takarabio.com
e-mail: ordersUS@takarabio.com	e-mail: technical_support@takarabio.com

Takara Bio USA, Inc.

United States/Canada: +1.800.662.2566 • Asia Pacific: +1.650.919.7300 • Europe: +33.(0)1.3904.6880 • Japan: +81.(0)77.565.6999

06.19 US