gCMAPWeb : a web interface for gene-set enrichment analysis

Thomas Sandmann

May 3, 2016

This document provides technical detail on the use of the gCMAPWeb package, including a short description of input and output data and the configuration of the web application. For expanded examples using real biological datasets available from public databases, please refer to the *tutorial* vignette instead.

Contents

1 Introduction

The gCMAPWeb package provides a graphical user interface for thegCMAP package, offering users a simple and comfortable way to compare gene sets or differential gene expression profiles to reference datasets through their web browser.

Leveraging the Rook package, gCMAPWeb can directly use R's built-in webserver to provide a single-user interface. Alternatively, gCMAPWeb can be registered with an apache webserver through the rApache module, offering a production-quality, multi-user application.

This vignette provides a step-by-step demonstration of

- the different types of analyses supported by gCMAPWeb
- its configuration and customization
- the deployment of gCMAPWeb instances through an apache webserver

2 Quickstart

To start gCMAPWeb on the local machine, start R, load the gCMAPWeb library and simply type `gCMAPWeb()`.

```
> library( gCMAPWeb )
> gCMAPWeb()
```

Your web browser will open the gCMAPWeb index page, prompting you to choose one of the tree supported query types. This gCMAPWeb instance is populated with small, simulated datasets, allowing you to get a first glimpse of the input / output of the gCMAPWeb package.

Many elements on the index.rhtml page can be configured through global options. Please refer to section "Customizing the gCMAPWeb web interface" for details.

3 Submitting gene sets and profiles

gCMAPWeb supports three different types of queries:

Directional queries

A list with two components, identifying up- and down-regulated genes.

For this query, up- and down-regulated gene sets are submitted separately, allowing gCMAPWeb to retrieve experiments in which the query genes changed expression in a consistent fashion, e.g. in the same (correlated) or opposite (anti-correlated) directions as your query. To identify reference experiments with significantly similar expression changes, gCMAPWeb calculates the JG score and obtains a parametric p-value based on a normal distribution. P-values are converted into local false-discovery rates using Benjamini Hochberg's multiple-testing adjustment method.

This query type should also be used if only either up- or down-regulated genes are known, by submitting only one of the two gene sets. Also, this query is the right choice if the set is expected to show consistent but unknown behavior. gCMAPWeb will retrieve reference datasets showing correlated and anti-correlated or results.

Non-directional queries

A single list of gene identifiers, potentially including both up- and down-regulated members.

This query is recommended if the query genes are expected to show mixed differential expression, e.g. that some members will be up- and others down-regulated. Using Fisher's exact test, gCMAPWeb will retrieve all experiments, in which a significant fraction of your genes of interest showed differential expression either way. P-values are converted into local false-discovery rates using Benjamini Hochberg's multiple-testing adjustment method.

Profile queries

a vector of differential expression scores (e.g. z-scores, assumed to be normally distributed) for all assayed genes, e.g. the complete results of a two-class differential expression analysis.

If you the complete set of differential expression scores (e.g. z-scores) from a global differential gene expression analysis, a profile query can be performed. Technically, a profile query is the reverse of a directional query: in the former, you provide the global differential gene expression scores and gCMAPWeb matches them to the gene identifiers in the reference database. In the latter, you provide a list of gene identifiers and gCMAPWeb retrieves the differential expression scores these genes received in other experiments from its reference database. As for the directional query type, gCMAPWeb calculates the JG score and derives parametric p-values using a normal distribution. For this procedure to be valid, the submitted scores should be approximately normally distributed.

For this query, two pieces of information need to be submitted for every gene:

- a gene identifier
- the associated differential gene expression score for this gene

Selecting reference datasets

Users can select either a single or multiple reference datasets from the same species for analysis. Internally, each set will be processed separately and p-value correction is applied within each reference dataset.

For more information on the elements in the user interface, please consult the help.html file available through the "Help" menu item on the gCMAPWeb index page.

4 Understanding gCMAPWeb results

Once you submitted your query, the gCMAP tool will search all selected databases for significantly similar experiments. The first result page will present a list of the most significant reference datasets matching your query (if any). A separate panel with results will be generated with results for each searched reference database.

Gene set reports

The main gCMAPWeb result page presents information about the detected similar experiments (not individual genes). At the top of the page, a density plot provides a high-level overview (Figure **??**). Reference instances with similarity scores >3 or <-3 are highlighted as green and blue dashes at the bottom of the plot and in the heatmap on the right.

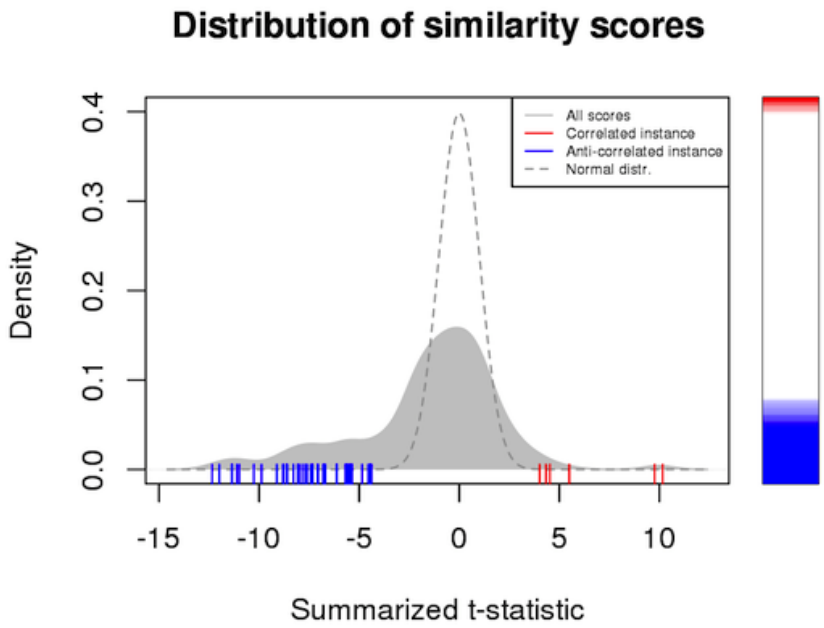

Figure 1: Example of a **gene-set score** density plot, as found on the main gCMAPWeb result page. The grey density plot summarizes the scores for all experiments in this reference dataset - similar or not. For reference, the normal distribution is shown as a dashed line. Reference instances with similarity scores >3 (correlated experiments) or <-3 (anti-correlated experiments) are indicated in the rug plot. On the right, the same similarity scores are shown as an ordered heatmap. High scores are shown at the top (green) and low scores at the bottom (blue).

In this example, only few experiments received high scores (green), indicating expression changes in the same direction as in the query, but a large number of experiments showed consistent changes of the query genes in the opposite direction (blue) than specified in the query.

Gene-level reports

For each reported gene set, gCMAPWeb generates a separate html page with detailed results for individual query genes. (If you submitted a complete profile, you will be presented with the scores for the those genes significantly changing in similar experiments.)

Gene-level results are linked via the nFound column in the main result table. For directional queries, the gene-level report will display the distribution of scores in a density plot.

While the density plot on the main result page displayed the similiarity scores for each reference dataset, summarizing a potentially large number of genes, this plot shows the differential gene expression scores of individual (query) genes (Figure **??**).

For more information about the elements presented in the gCMAPWeb output, please consult the help.html file available through the "Help" menu item on the gCMAPWeb index page for additional examples for all query types.

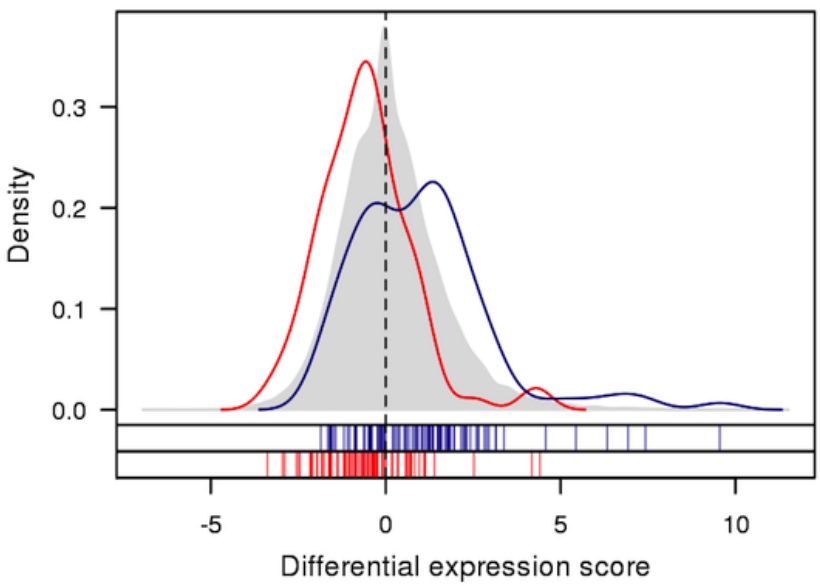

Figure 2: Example of a **gene score** density plot. The grey density plot summarizes the scores for all genes assayed in this reference dataset. For reference, the normal distribution is shown as a dashed line. The distribution of query genes submitted as "up-regulated" is shown in green, that of "down-regulated" query genes is shown in blue. (For non-directional queries, a single query gene density is shown in black.) Query gene scores are also indicated in the rug plot following the same color scheme as outlined above. In this example, an **anti-correlated result** is shown: query genes submitted as "up-regulated" (green) are shifted toward negative scores, while "down-regulated" query genes are shifted to positive scores.

5 Configuration

This section will take you through the steps necessary to search your own datasets with gCMAPWeb.

Reference datasets

gCMAPWeb can query either quantitative differential expression data, e.g. z-scores, stored in `NChannelSet` objects, or gene-set collections provided as `CMAPCollection` objects. To avoid reading large datasets fully in to memory, gCMAPWeb can take advantage of the bigmemory and bigmemoryExtras packages to retrieve only the required data from a binary file stored on disk. Note: At the time or writing these two packages were only available for Unix and Mac OS X but not Windows operating systems.

All reference datasets must be provided with **Entrez gene identifiers**. If available, information from the `abstract` , `title` slots will be used for labels and pop-over help on the submission page (see figure **??**).

For example, please take a look at the `cmap1 NChannelSet` and the `cmap5 CMAPCollection` objects provided with the package.

```
> library(gCMAPWeb)
> data( "cmap1" )
> cmap1
```

```
NChannelSet (storageMode: lockedEnvironment)
assayData: 1000 features, 10 samples
```

```
  element names: p, z
protocolData: none
phenoData
  sampleNames: Exp1 Exp2 ... Exp10 (10 total)
  varLabels: Name Treatment Date
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation: org.Hs.eg
```

```
> experimentData( cmap1 )@title
```

```
[1] "Reference dataset 1"
```

```
> abstract( cmap1 )
```

```
[1] "A first test dataset with experiment with random z-scores for 1000 genes"
```

```
> data( "cmap5" )
> cmap5
```

```
CMAPCollection (storageMode: lockedEnvironment)
assayData: 1000 features, 10 samples
  element names: members
protocolData: none
phenoData
  sampleNames: Exp1 Exp2 ... Exp10 (10 total)
  varLabels: Name signed
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

By default, `NChannelSet` reference objects are made available for all three query types, unsigned, directional and profile. `CMAPCollections`, on the other hand, only provide gene set membership information. If gene-signs are included in the `CMAPCollection`, indicating whether genes are expected to be up- or down-regulated, the reference dataset is made available for non-directional (unsigned) and profile queries. If the CMAPCollection contains (any) unsigned gene sets, the reference dataset can only be selected for non-directional queries. To manually include or exclude reference datasets from specific query types, a character vector with the supported query types (unsigned, directional, profile) can be specified by specifying `experimentData@other$supported.query` in the `NChannelSet` or `CMAPCollection`.

To hide the cmap1 reference dataset on the profile submission page, simply include information about the supported query type(s) in the `experimentData` slot

```
> cmap1@experimentData@other$supported.query <- c("unsigned", "directional")
```

By default, all of the sample annotations available in the `phenoData` slot of the reference dataset will be included in the table of significantly similar reference experiments. To exclude non-informative columns, an additional `include` column can be included as an additional `varMetdata` column, indicating for each `phenoData` column whether is it should be included in the result table or not.

For example, the following commands prevents cmap1's "Date" column from the being displayed.

```
> head( pData( cmap1 ), n=3)
```

```
     Name Treatment       Date
Exp1 Exp1         a 01.01.2010
Exp2 Exp2         b 01.02.2010
Exp3 Exp3         c 01.03.2010

> varMetadata( cmap1 )

            labelDescription channel
Name         Experiment name    <NA>
Treatment              Agent    <NA>
Date      Date of experiment    <NA>

> varMetadata( cmap1 )$include <- c(TRUE, TRUE, FALSE)
> varMetadata( cmap1 )

            labelDescription channel include
Name         Experiment name    <NA>    TRUE
Treatment              Agent    <NA>    TRUE
Date      Date of experiment    <NA>   FALSE
```

Reference datasets are specified by providing the full path to the corresponding Rdata file in the configuration file. (Please note that each Rdata file should only contain a single eSet.)

Essential information : the configuration file

gCMAPWeb's configuration file is in `YAML` format provides details on the reference datasets to be included in gCMAPWeb in the form of a nested list. It includes e.g. the path to the RData file and the name of the associated annotation package.

Here is a simple example of a configuration for a gCMAPWeb instance supporting only human gene queries:

```
species:
  human:
    annotation: org.Hs.eg
    platforms:
     - hgug4100a
     - hgug4110b
    cmaps:
      reference1: /home/data/ref1.rdata
      reference2: /home/data/ref2.rdata
```

In this example

- species: contains only one supported species, human.
- annotation: specifies that mappings between gene identifiers are retrieved from Bioconductor's `org.Hs.eg` annotation package. gCMAPWeb automatically loads the specified annotation packages, so they must be installed on the user's system.
- platforms: specifies two microarrays (hgug4100a, hgug4110b). Users submitting queries with "probe" as identifier type will be prompted to choose one of these two supported platforms. gCMAPWeb automatically loads the specified annotation packages, so they must be installed on the user's system.

- cmaps: contains the full path to the reference datasets for this species. Each path must be preceded by a unique identifier (e.g. reference.1). As this identifier is used as an object name both in R and javascript, it must be a single alpha-numeric single string and must not contain spaces, hyphens, dots, etc. This identifier is only displayed on the submission page if the eSet does not have a title.

Additional species can be added by duplicating the `species` block of the configuration and modifying the respective fields. (See default configuration file for an example with two species.)

When gCMAPWeb is invoked without additional parameters, the default configuration file in the `config` directory of the gCMAPWeb package is read. You can obtain the full path to the default configuration file on your system with the following command:

```
> system.file("config", "config.yml", package = "gCMAPWeb")
```

To read your own, customized configuration file instead, provide its path via the `config.file.path` parameter.

```
> gCMAPWeb( config.file.path = "/path/to/your/config_file.yml")
```

Additional information : eSet slots

The configuration file provides all required information to start a gCMAPWeb instance. To fine-tune the information displayed about each reference dataset, three different settings can be customized on the object level using the default slots of eSet objects.

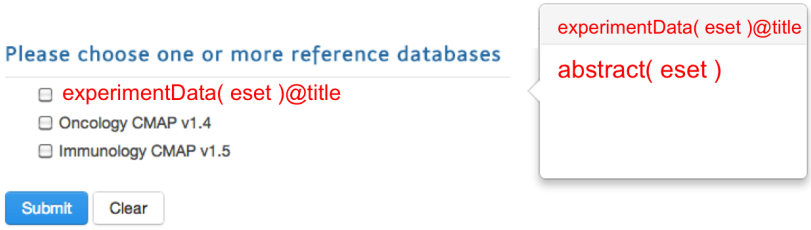

Figure 3: Reference dataset information

If present, three different slots are used by gCMAPWeb (see figure **??**):

- abstract: Text in the abstract slots will be displayed as pop-up information upon mouse-over. If no abstract is provided, generic information about the number of experiments in the dataset is displayed instead.
- title: The eSet title will be used as the reference database name displayed on the submission page. If not title is provided, the unique identifier specified for this dataset in the configuration file is used instead.
- other$supported.query: Defines for which query type this dataset should be made available, one or more of "directional", "unsigned" and /or "profile". If no other$supported.query information is provided, the class of the reference dataset evaluated instead with the following defaults:
  - NChannelSet: available for all three query types
  - CMAPCollection, unsignedavailable for non-directional/unsigned queries only
  - CMAPCollection, signedavailable for non-directional/unsigned and profile queries

6 Customizing the gCMAPWeb web interface

Many elements of the user interface and parameters of the gCMAPWeb search methods can be set through global parameters. These can either be called before executing the `gCMAPWeb()` function for local instances or be included in the start-up script executed by the rApache server (see below).

Navigation bar

Please note that text elements displayed on the html page(s) are intepreted as html code.

- site.title Brand shown on the upper left of the menu bar
- home.url URL linked to brand item in the menu bar
- doc.url URL linked to "Help" item in the menu bar
- feedback.url URL linked to "Feedback" item in the menu bar
- contact.email URL linked to "Contact" item in the menu bar
- name.out additional element on the far right of the menu bar
- link.out URL linked to "name.out" item in the menu bar

Processing options

- save.intermediates Logical, should intermediate files be saved (for debugging). Default: TRUE
- element AssayDataElementName of the assayData slot with differential expression scores to retrieve from `NChannelSet` reference datasets. Default: "z"
- min.set.size Minimum number of elements a reference gene set must contain to be searched. (Note: not to be confused with option min.found, the minimum number of overlap between query and reference sets) Default: 5
- max.set.size Maximum number of elements a reference gene set may contain to be searched. (Note: only applies to non-directional and profile queries) Default: Inf
- lower.threshold Lower score threshold applied to reference datasets or profile queries to identify signficantly down-regulated genes. Default=-3
- higher.threshold Lower score threshold applied to reference datasets or profile queries to identify signficantly up-regulated genes. Default= 3
- induce.from.element AssayDataElementName of the assayData slot with differential expression scores to threshold when gene sets are induced from `NChannelSet` reference datasets. Default: "z"
- cmaps.concatenated.by Character string used to concatenate multiple requested reference dataset names in the html POST request . Default: ","
- max.results Maximum number of similar reference datasets to return with FDR < option max.padj. As gene-level pages are created for each significant set, increasing this number can lead to increased processing time. Default: "50"
- min.found Minimum number of query genes found in a reference gene set for it to be included in the result table. Default: "1"
- max.padj Maximum adjusted p-value / FDR for a gene set to be included in the result table. Default: 0.1

Output options

- gene.level.report Logical, should sub-pages be created for each of the top max.results significant gene sets ? Default=TRUE
- gene.level.plot Logical, should gene-level pages include a plot of score distributions ? Default=TRUE
- show.heatmap Logical, should gene-level scores (if available) be displayed in a heatmap on the main result page ? Default=TRUE
- excluded.cols Character vector with columns to exclude from a CMAPResults object when the html output is created. Default=c("geneScores", "signed", "pval", "UID", "z.shift", "log_fc.shift", "mod_fc.shift")
- swap.colnames List of column names to rename when the html output is created. Default=list(padj="FDR", nFound="Genes")
- table.javascript Logica, should the dataTable javascript module be used to render the output html tabels ? Default=TRUE

show.heatmap

Index page

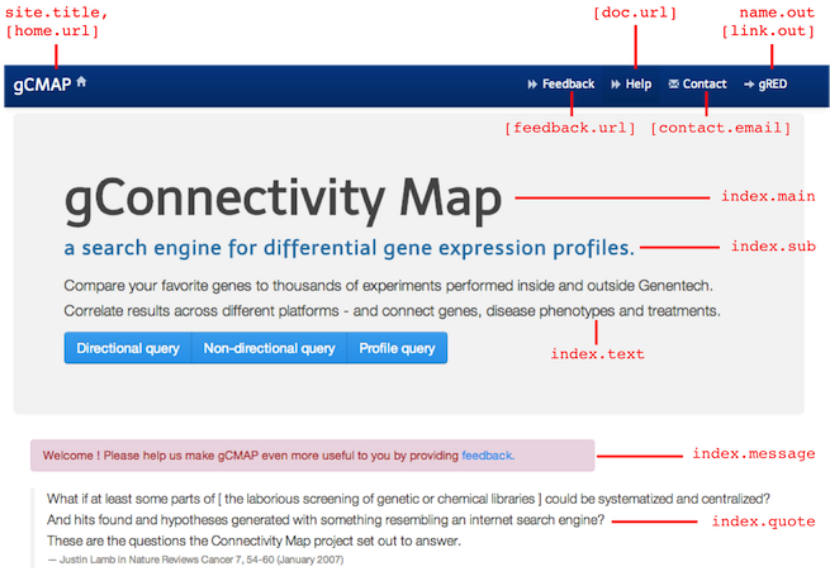

Figure 4: Options available to customize the index page

The following options are available to customize the index html page (see figure figure **??** for a graphical overview). Please note that text elements displayed on the html page(s) are intepreted as html code.

- index.main Main title shown on the index page. Default="gConnectivity Map"
- index.sub Subtitle shown on the index page. Default="a search engine for differential gene expression profiles."

- index.text Text shown on the the index page. Default=”Compare your favorite genes to a reference database of differential expression experiments”
- index.message Text shown in a box highlighed in green on the the index page. Default=NULL
- index.quote Text shown as quote on the the index page.
- supported.inputType They query type(s) offered. Default: c(”single”, ”non-directional”, ”directional”, ”profile”)
- supported.idType The supported gene identfier types. Default: c(”symbol”, ”entrez”, ”probe”)

Examples

The following options specify which gene identifiers are pasted into the submission boxes when the ”Example query” button is pressed.

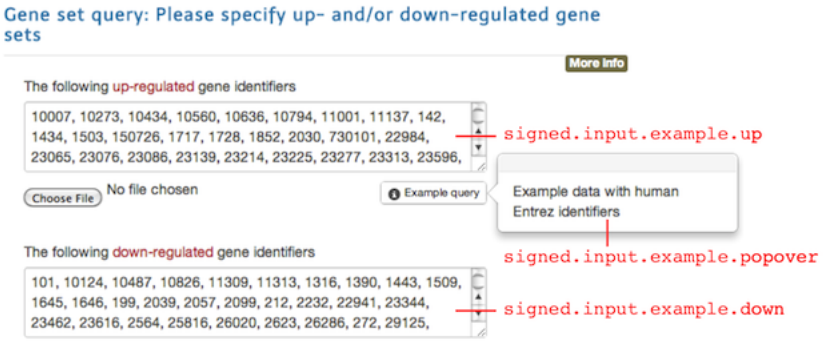

Figure 5: Options populating the Example query button on the directional query submission page.

The following options are available to customize the index html page (see figure figure **??** for a screenshot of the directional query submission page with relevant options).

- single.gene.example Example input for a single gene lookup.
- single.gene.example.popover Text to be displayed in the pop-over element for the Example query button for singel gene lookups.
- signed.input.example.down Example query for down-regulated genes in a directional query
- signed.input.example.up Example query for up-regulated genes in a directional query
- signed.input.example.popover Text to be displayed in the pop-over element for the Example query button for signed queries.
- unsigned.input.example Example query for a non-directional query
- unsigned.input.example.popover Text to be displayed in the pop-over element for the Example query button for unsigned queries.
- profile.input.example Example gene identifier / score pairs for a non-directional query. Newlines have to be included as `\\n` strings between consecutive id-score pairs.
- profile.input.example.popover Text to be displayed in the pop-over element for the Example query button for profile queries.

Figure legends

- gene.profile.legend Legend for the density plot on the main result page for profile queries
- gene.set.legend Legend for the density plot on the main result page for directional and non-directional queries
- heatmap.legend Legend for the heatmap plot on the main result page for directional and non-directional queries
- gene.density.legend Legend for the density plot on gene-level report pages
- gene.pie.legend Legend for the pie charts on gene-level report pages

For example, the main title of the gCMAPWeb index page can be retrieved and set using the `site.title` option.

```
> getOption( "site.title", default="gCMAP")
> options( site.title="New site title")
```

7 Deploying gCMAPWeb through rApache

To run gCMAPWeb in a multi-user environment, the application can be deployed through rApache. For installation and configuration options of Apache and rApache, please consult the respective project pages. The following instructions assume that you have access to the Apache installation directory, especially the `httpd.conf` file.

Once a working webserver is available, the following steps are required to deploy gCMAPWeb:

1. Locate the the `htdocs` directory within the gCMAPWeb installation directory on your system and copy its contents (including all subdirectories) into the htdocs directory of your Apache installation. You can retrieve the location of gCMAPWeb's htdocs directory by issuing the following commands in your R console:

```
> library( gCMAPWeb )
> system.file("htdocs", package="gCMAPWeb")
```

2. Locate the `gCMAP_app.R` , `rapache_config.R` and `config.yml` files in gCMAPWeb's installation directory and copy them to a location accessible to the Apache server.

```
> system.file("config", "rapache", "gCMAP_app.R", package="gCMAPWeb")
> system.file("config", "rapache", "rapache_config.R", package="gCMAPWeb")
> system.file("config", "config.yml", package="gCMAPWeb")
```

3. Edit the new copy of the `gCMAP_app.R text file with a text editor of your choice and change the 'root.url' variable (first line) to point toward the location of your Apache htdocs directory.`

4. Edit the new copy of the `rapache_config.R` text file with a text editor of your choice and change the `config.file.path` variable to point toward the location of your copy of the `config.yml` file. The the `rapache_config.R` is executed upon startup for each R session by the Apache web server. Use the `options` command to set any global options to fine-tune the look and behavior of gCMAPWeb

5. Open the `httpd.conf` of your Apache webserver and ensure that rApache has been installed correctly, e.g. by testing that the `r-info` test application is accessible, as described in the rApache manual.

6. Add or modify the `REvalOnstartup` and `RSourceOnStartup` command lines to include the following lines (replace `PATH_TO_YOUR` with the path to your `rapache_config.R` file):

```
REvalOnStartup "library(gCMAPWeb)"
RSourceOnStartup "PATH_TO_YOUR/rapache_config.R"
```

   These lines will instruct Apache to load the gCMAPWeb package upon starting an R session and execute the commands in the `rapache_config.R` R script. (You can add additional commands to the `rapache_config.R` file, e.g. specifying the number of cores available on your system, etc.)

7. To register the gCMAPWeb application with the Apache webserver, add the following lines to the `httpd.conf` , replacing `PATH_TO_YOUR` with the path to your `gCMAP_app.R` file:

```
<Location /gcmap>
        SetHandler r-handler
        RFileEval PATH_TO_YOUR/gCMAP_app.R:Rook::Server$call(gcmap)
</Location>
```

8. After restarting the Apache server, your gCMAPWeb application will be available at

```
YOUR_HOST_NAME://gcmap/index.rhtml
```

```
> sessionInfo()

R version 3.3.0 RC (2016-04-26 r70550)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] gCMAPWeb_1.12.0      Rook_1.1-1           gCMAP_1.16.0
 [4] limma_3.28.0         GSEABase_1.34.0      graph_1.50.0
 [7] annotate_1.50.0      XML_3.98-1.4         AnnotationDbi_1.34.0
[10] IRanges_2.6.0        S4Vectors_0.10.0     Biobase_2.32.0
[13] BiocGenerics_0.18.0

loaded via a namespace (and not attached):
 [1] splines_3.3.0      GSEAlm_1.32.0      xtable_1.8-2       lattice_0.20-33
 [5] brew_1.0-6         Category_2.38.0    DESeq_1.24.0       hwriter_1.3.2
 [9] tools_3.3.0        grid_3.3.0         DBI_0.4            genefilter_1.54.0
[13] yaml_2.1.13        survival_2.39-2    RBGL_1.48.0        Matrix_1.2-6
[17] geneplotter_1.50.0 RColorBrewer_1.1-2 RSQLite_1.0.0
```