

Recreating EnrichmentMap tutorials with RCy3

Julia Gustavsen

May 24, 2016

Contents

1 Purpose:	1
2 Enrichment Map	1
2.1 Functional enrichment analysis	1
2.2 Reproducible Functional enrichment analysis	2
3 GSEA processed data	2
3.1 Load the appropriate libraries	2
3.2 Important note	2
3.3 Load functions for creating Enrichment map	2
3.4 Description of files that can be used	3
3.5 Send data to the cytoscape network	4
3.6 Set the parameters for use in the Enrichment Map.	4
3.7 Save Enrichment map network image	4
3.8 Change the layout of the network	6
4 Summarizing Enrichment Results with Enrichment Maps	8
4.1 Recreating Protocol 4 - Summarize Enrichment Results with Enrichment Maps	8
4.1.1 Option 1: Load enrichment results from g:Profiler	8
4.2 Helpful references:	10

1 Purpose:

- Recreating tutorials from Bader lab using Rcy3 and cytoscape.
- Create and test out functions using RCy3 that make Enrichment Map easy to use in R.

2 Enrichment Map

2.1 Functional enrichment analysis

Many scientists use perform experiments to determine which biological pathways are expressed more in certain diseases or conditions. Using processed sequence data from RNAseq experiments, different treatments and/or samples can be visualized for genes that are more or less enriched compared to the baseline. This informs which genes are important for regulating processes related to a disease or a treatment. Visualizing

which genes are statistically more highly expressed under certain conditions can be helpful for interpretation and further analysis of the data and it can help determine which pathways are present during the specific disease or treatment.

The functional enrichment analysis is done outside of this vignette. We will use already processed data and make a network from these data in Cytoscape using RCy3.

2.2 Reproducible Functional enrichment analysis

The basic workflow is using R scripts with RCy3 to build EnrichmentMaps that can be visualized and analysed in Cytoscape. The benefit of using R scripts is that it is easier to reproduce your analysis or to repeat it with different data.

3 GSEA processed data

So what we will do today is to use data already processed in Gene Set Enrichment Analysis (GSEA which “determines whether an *a priori* defined set of genes shows statistically significant, concordant differences between two biological states”).

We will use this processed data to make an Enrichment Map in Cytoscape from R.

3.1 Load the appropriate libraries

```
library(RCy3)
library(httr)
library(RJSONIO)
```

3.2 Important note

- Make sure Cytoscape is open before running the code below!

3.3 Load functions for creating Enrichment map

```
source("./functions_to_add_to_RCy3/working_with_EM.R")
```

Create the connection from R to Cytoscape

```
# first, delete existing windows to save memory:
deleteAllWindows(CytoscapeConnection())
cy <- CytoscapeConnection ()
```

Examine the commands that are available in Enrichment Map.

```
getEnrichmentMapCommandsNames(cy, "build")
```

```
## [1] "analysisType"      "classDataset1"      "classDataset2"
## [4] "coefficients"      "enrichments2Dataset1" "enrichments2Dataset2"
## [7] "enrichmentsDataset1" "enrichmentsDataset2" "expressionDataset1"
## [10] "expressionDataset2" "gmtFile"            "phenotype1Dataset1"
## [13] "phenotype1Dataset2" "phenotype2Dataset1" "phenotype2Dataset2"
## [16] "pvalue"            "qvalue"             "ranksDataset1"
## [19] "ranksDataset2"      "similaritycutoff"
```

```
getEnrichmentMapCommandsNames(cy, "gseabuild")
```

```
## [1] "combinedconstant" "edbdir"             "edbdir2"
## [4] "expressionfile"   "expressionfile2"    "overlap"
## [7] "pvalue"           "qvalue"             "similaritymetric"
```

3.4 Description of files that can be used

See the Bader lab website for full explanation: https://github.com/BaderLab/EnrichmentMapApp/blob/EM_Cyto3_port/EnrichmentMapPlugin/doc/EM_wiki_manual.txt

- **“gmtFile”**: Tab-separated file where the header has name, description and samples and each row is one gene in the geneset.
- **“analysisType”** Analysis type can be “generic”, “GSEA”, or “DAVID/BiNGO/Great”
- **“classDataset1”** Classes (optional) for dataset1
- **“classDataset2”** Classes (optional) for dataset2
- **“coefficients”** Similarity Coefficient type (contains a typo in the name, but you must type it that way). Can be “OVERLAP”, “JACCARD”, or “COMBINED” (both OVERLAP and JACCARD)
- **“similaritycutoff”** Similarity Cutoff for the coefficient chosen, 0.0 is least similar and 1.0 is most similar.
- **“enrichments2Dataset1”** Tab-separated File containing gene names and their p-values from enrichment condition 2 from dataset 1
- **“enrichments2Dataset2”** Tab-separated File containing gene names and their p-values from enrichment condition 2 from dataset 2
- **“enrichmentsDataset1”** Tab-separated File containing gene names and their p-values from enrichment condition 1 from dataset 1
- **“enrichmentsDataset2”** Tab-separated File containing gene names and their p-values from enrichment condition 1 from dataset 2
- **“expressionDataset1”** Expression (optional) tab-separated file containing gene name, description and expression values. For dataset1
- **“expressionDataset2”** Expression (optional) tab-separated file containing gene name, description and expression values. For dataset2
- **“phenotype1Dataset1”** Default is “Up” for Phenotype1. Can change to a descriptive word for enrichment 1
- **“phenotype2Dataset1”** Default is “Down” for Phenotype2. Can change to a descriptive word for enrichment 2
- **“phenotype1Dataset2”** same as above but for Dataset2
- **“phenotype2Dataset2”** same as above but for Dataset2
- **“pvalue”** P-value Cutoff for enrichment data to be used.
- **“qvalue”** False discovery rate (Q-value) cutoff. Used with multiple comparisons.
- **“ranksDataset1”** Ranks (optional) tab-separated file containing genes and their rank (from GSEA) for Dataset1
- **“ranksDataset2”** Ranks (optional) tab-separated file containing genes and their rank (from GSEA) for Dataset2

3.5 Send data to the cytoscape network

So first we read in the data from the supplied files and set the parameters.

Note on file paths: You cannot use relative paths when constructing EnrichmentMaps from R. The filenames need to be given as their absolute paths.

```
path_to_file <- "/home/julia_g/windows_school/gsoc/EM-tutorials-docker/notebooks/data/"

enr_file <- paste0(path_to_file,
                  "gprofiler_results_mesenonly_ordered_computedinR.txt")
exp_file <- paste0(path_to_file,
                  "MesenchymalvsImmunoreactive_expression.txt")
```

3.6 Set the parameters for use in the Enrichment Map.

```
em_params <- list(analysisType = "generic",
                 enrichmentsDataset1 = enr_file,
                 pvalue = "1.0",
                 qvalue = "0.00001",
                 expressionDataset1 = exp_file,
                 similaritycutoff = "0.25",
                 coeffecients = "JACCARD")
```

No graph details are returned, this is just setting the parameters that will be sent to Cytoscape via RCy3.

Now build the enrichment map

```
EM_1 <- setEnrichmentMapProperties(cy,
                                "build",
                                em_params)
```

```
## [1] "Successfully built the EnrichmentMap."
## [1] "Cytoscape window EM1_Enrichment Map successfully connected to R session."
```

These parameters can also be set in Cytoscape, but we are setting them here via script. The function `setEnrichmentMapProperties()` also attaches the window created in Cytoscape to our R session, so that we are able to manipulate the stylistic aspects of our network by using “EM_1”.

3.7 Save Enrichment map network image

```
fitContent(EM_1)
Sys.sleep(10)
saveImage(EM_1,
          "EM_1",
          "png",
          h = 2000)
```


3.8 Change the layout of the network

Can change any of the visual properties of “EM_1” now. For demonstration let’s change the layout.

```
layoutNetwork(EM_1,
              'kamada-kawai')
Sys.sleep(5)
## save network visualization
saveImage(EM_1,
          "EM_1_kamada-kawai",
          "png",
          h = 2000)
```

We can also save our network file to use at a later date or to send to collaborators.

```
saveNetwork(EM_1,
            "EM_1") ## Creates "EM_1.cys" which can be reopened in Cytoscape
```

In our R session we are connected to the graph in Cytoscape and can manipulate the graph’s visual properties, but we do not have the graph information in R.

```
EM_1@graph
```

```
## A graphNEL graph with directed edges
## Number of Nodes = 0
## Number of Edges = 0
```

If we want the graph data pulled in to R then we can set the argument “copy.graph.to.R” to TRUE in `setEnrichmentMapProperties()`. This pulls in our EM analysis graph to Cytoscape and stores it as EM_1_2.

```
EM_1_2 <- setEnrichmentMapProperties(cy,
                                   "build",
                                   em_params,
                                   copy.graph.to.R = TRUE)
```

```
## [1] "Successfully built the EnrichmentMap."
## [1] "Cytoscape windowEM2_Enrichment Map successfully connected to R session and graph copied to R."
```

Now if we have a look at the graph part of the stored object we will see we have retrieved the information from the graph.

```
EM_1_2@graph
```

```
## A graphNEL graph with directed edges
## Number of Nodes = 297
## Number of Edges = 1793
```

```
print(noa.names(getGraph(EM_1_2))) # retrieves node attributes from the graph
```

```
## [1] "name" "EM2_GS_DESCR"
## [3] "EM2_Formatted_name" "EM2_Name"
## [5] "EM2_GS_Source" "EM2_GS_Type"
## [7] "EM2_pvalue_dataset1" "EM2_Colouring_dataset1"
## [9] "EM2_fdr_qvalue_dataset1" "EM2_gs_size_dataset1"
```

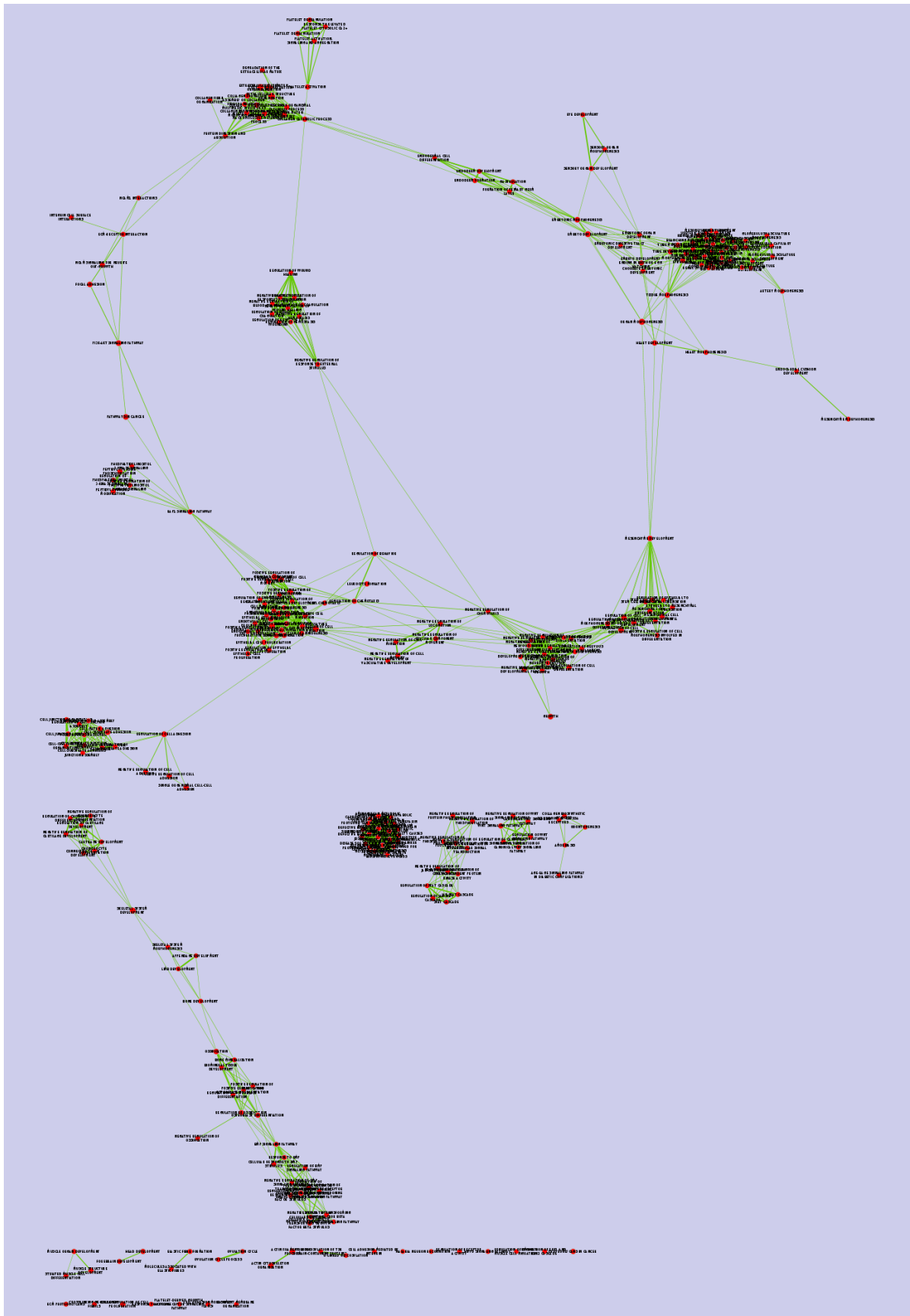


Figure 2:
7

4 Summarizing Enrichment Results with Enrichment Maps

4.1 Recreating Protocol 4 - Summarize Enrichment Results with Enrichment Maps

4.1.1 Option 1: Load enrichment results from g:Profiler

Load in the datafiles

```
path_to_file <- "/home/julia_g/windows_school/gsoc/EM-tutorials-docker/notebooks/data/"

enr_file <- paste0(path_to_file,
                  "gprofiler_results_mesenonly_ordered.txt")

exp_file <- paste0(path_to_file,
                  "MesenchymalvsImmunoreactive_expression.txt") # this one works.
#expression_RNA_seq <- paste0(path_to_file,
#                             "MesenchymalvsImmunoreactive_RNSseq_expression_ed.txt") ## not working
# ranks_file <- paste0(path_to_file,
#                      "MesenchymalvsImmunoreactive_RNA_seq_ranks.rnk") ## does not work
ranks_file <- paste0(path_to_file,
                    "MesenchymalvsImmunoreactive_edger_ranks.rnk")

classes_file <- paste0(path_to_file,
                      "MesenchymalvsImmunoreactive_RNAseq_classes.cls")
```

```
em_params <- list(analysisType = "generic",
                 enrichmentsDataset1 = enr_file,
                 pvalue = "1.0",
                 qvalue = "0.0001",
                 expressionDataset1 = exp_file,
                 ranksDataset1 = ranks_file, ## not working from RNA seq data
                 classDataset1 = classes_file,
                 phenotype1Dataset1 = "Mesenchymal", # shows up as positive, red, nodes.
                 phenotype2Dataset1 = "Immunoreactive",
                 similaritycutoff = "0.25",
                 coeffecients = "JACCARD")
```

```
EM_ex_4 <- setEnrichmentMapProperties(cy,
                                    "build",
                                    em_params)
```

```
## [1] "Successfully built the EnrichmentMap."
## [1] "Cytoscape window EM3_Enrichment Map successfully connected to R session."
```

```
fitContent(EM_ex_4)
Sys.sleep(10)
saveImage(EM_ex_4,
          "EM_ex_4",
          "png",
          h = 2000)
```

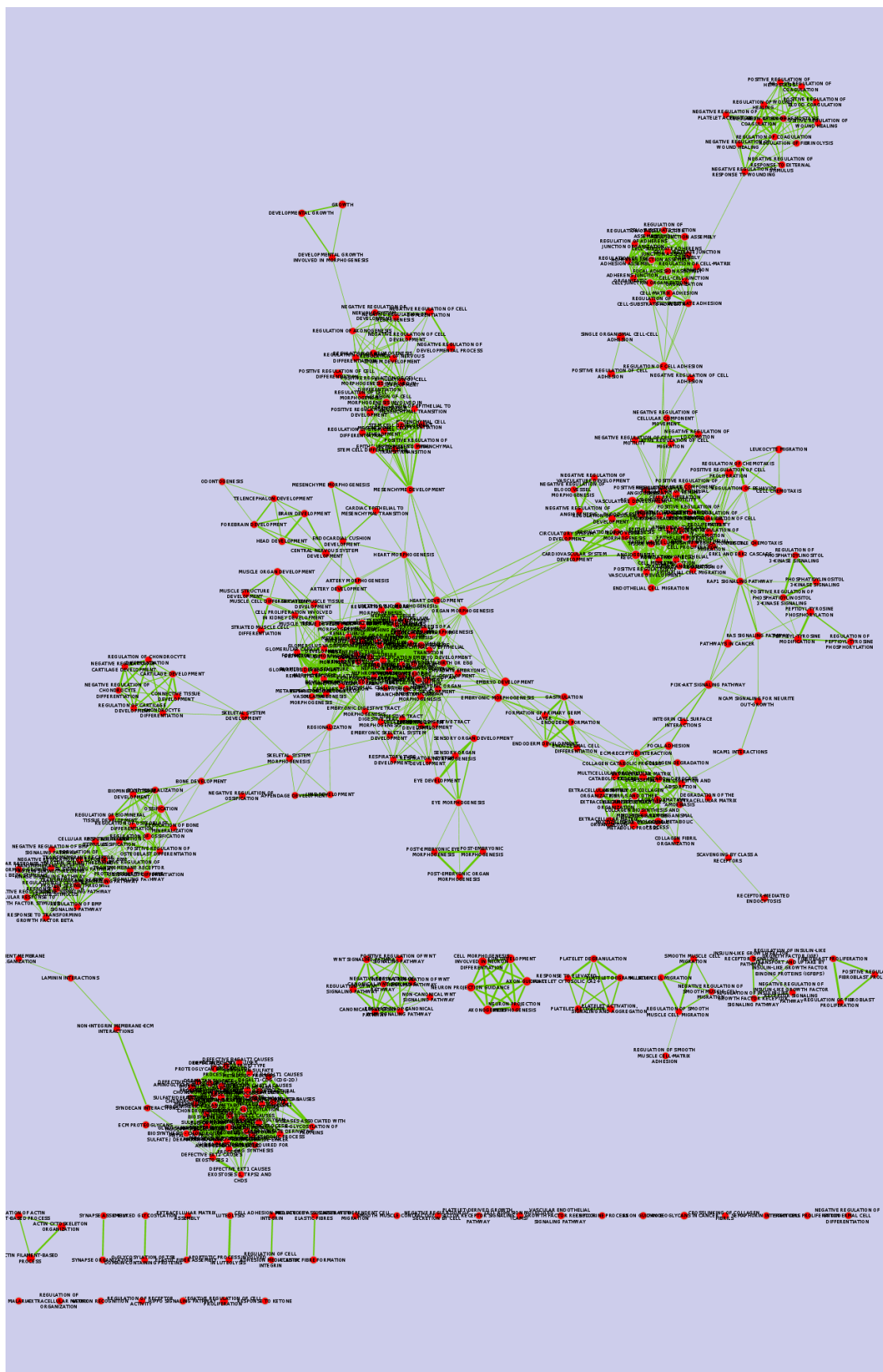



Figure 3:
9

4.2 Helpful references:

Reference for the API. This site describes the cyREST API: <http://idekerlab.github.io/cyREST/>