WebGestaltR is the R version of our well-known web application tool WebGestalt (www.webgestalt.org) that has been visited 57,880 times by 26,233 users from 140 countries and territories in 2016 and has also been cited 371 in 2016. The advantage of this R package is it can be easily integrated to other pipeline or simultaneous analyze multiple gene lists.

WebGestaltR function can perform two popular enrichment analyses: ORA (Over-Representation Analysis) and GSEA (Gene Set Enrichment Analysis). Based on the user uploaded gene list or gene list with scores (for GSEA method), WebGestaltR function will first map the gene list to the entrez gene ids and then summary the gene list based on the GO (Gene Ontology) Slim. After performing the enrichment analysis, WebGestaltR function also returns an user-friendly HTML report containing the ID mapping table, GO Slim summary result and the enrichment analysis result. If the functional categories have the DAG (directed acyclic graph) structure, the structure of the enriched categories can also be visualized in the report.

Manual of WebGestaltR

Jing Wang

March 7, 2017

1 Introduction

2 Environment

WebGestaltR requires R version 3.3 or later, which can be downloaded from the website http://www.r-project.org/. WebGestaltR package requires the following packages: pkgmaker (>=0.22), rjson (>=0.2.15), data.table (>=1.10.0), PythonInR (>=0.1-3), parallel (>3.3.2), doParallel (>1.0.10) and foreach (>1.4.0), which can be installed as follows.

```
>install.packages(”pkgmaker”)

>install.packages(”rjson”)

>install.packages(”data.table”)

>install.packages(”PythonInR”)

>install.packages(”parallel”)

>install.packages(”doParallel”)

>install.packages(”foreach”)
```

3 WebGestaltR

After building up the basic environment mentioned above, the users can install the WebGestaltR package and use it to analyze networks.

```
> library("WebGestaltR")
> #######ORA example#########
> interestGeneFile <- system.file("extdata","interestingGenes.txt",package="WebGestaltR")
> referenceGeneFile <- system.file("extdata","referenceGenes.txt",package="WebGestaltR")
> outputDirectory <- getwd()
> enrichResult <-WebGestaltR(enrichMethod="ORA", organism="hsapiens", enrichDatabase="geneontology_Bi

Uploading the functional categories...
Uploading the gene list...
Uploading the reference gene list...
Summary the uploaded gene list by GO Slim data...
Perform the enrichment analysis...
Generate the final report...
Results can be found in the /private/var/folders/7r/3czb3xbj4b59zwhxtzh_0h3w0000gn/T/RtmpLbEk5g/Rbuil


>
> ########GSEA example#######
> #geneRankFile <- system.file("extdata","GeneRankList.rnk",package="WebGestaltR")
> #outputDirectory <- getwd()
> #enrichResult <-WebGestaltR(enrichMethod="GSEA", organism="hsapiens", enrichDatabase="pathway_KEGG"
>
```

3.1 Input

This section describes the arguments of the WebGestaltR function:

1. *enrichMethod*: WebGestaltR supports two enrichment analysis methods: ORA (Over-Representation Analysis) and GSEA (Gene Set Enrichment Analysis).

2. *organism*: Currently, WebGestaltR supports 12 organisms. Users can use the function *listOrganism* to check the available organisms. Users can also input *others* to perform the enrichment analysis based on other organisms not supported by WebGestaltR. For the other organisms, users need to upload the enrichment categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above uploaded data should have the same ID type.

3. *enrichDatabase*: The functional categories for the enrichment analysis. Users can use the function *listGeneset* to check the available functional databases for the selected organism. Users can also input *others* to upload the functional database not supported by WebGestaltR for the selected organism.

4. *enrichDatabaseFile*: If users set *organism* as *others* or set *enrichDatabase* as *others*, users need to upload a GMT file as the functional categories for the enrichment analysis. The extension of the file should be *gmt* and the first column of the file is the category ID, the second one is the external link for the category. Genes annotated to the category are from the third column. All columns are separated by tab.

5. *enrichDatabaseType*: If users set *enrichDatabase* as *others*, WebGestaltR will also perform ID mapping for the uploaded GMT file. Thus, users need to set the ID type of the genes in the *enrichDatabaseFile*. If users set *organism* as *others*, users do not need to set this ID type because WebGestaltR will not perform ID mapping for other organisms. The supported ID type of the WebGestaltR for the selected organism can be found by the function *listIDType*.

6. *enrichDatabaseDescriptionFile*: Users can also upload a description file for the uploaded *enrichDatabaseFile*. The extension of the description file should be *des*. The description file contains two columns: the first column is the category ID that should be exactly the same as the category ID in the uploaded *enrichDatabaseFile* and the second column is the description of the category. All columns are separated by tab.

7. *interestGeneFile*: If *enrichMethod* is *ORA*, the extension of the *interestGeneFile* should be *txt* and the file can only contain one column: the interesting gene list. If *enrichMethod* is *GSEA*, the extension of the *interestGeneFile* can be *txt* or *rnk* and the file should contain two columns separated by tab: the gene list and the corresponding scores.

8. *interestGene*: Users can also use the R object as the input. If *enrichMethod* is *ORA*, *interestGene* should be an R *vector* object containing the interesting gene list. If *enrichMethod* is *GSEA*, *interestGene* should be an R *data.frame* object containing two columns: the gene list and the corresponding scores.

9. *interestGeneType*: The ID type of the interesting gene list. The supported ID type of the WebGestaltR for the selected organism can be found by the function *listIDType*. If the *organism* is *others*, users do not need to set this parameter.

10. *collapseMethod*: The method to collapse the duplicate ids for the GSEA method. *mean*, *median*, *min* and *max* represent the mean, median, minimum and maximum of scores for the duplicate ids.

11. *referenceGeneFile*: For ORA method, the users need to upload the reference gene list. The extension of the *referenceGeneFile* should be *txt* and the file can only contain one column: the reference gene list.

12. *referenceGene*: For ORA method, users can also use the R object as the reference gene list. *referenceGene* should be an R *vector* object containing the reference gene list.

13. *referenceGeneType*: The ID type of the reference gene list. The supported ID type of the WebGestaltR for the selected organism can be found by the function *listIDType*. If the *organism* is *others*, users do not need to set this parameter.

14. *referenceSet*: Users can directly select the reference set from the existing platform in the WebGestaltR

and do not need to upload the reference set. All existing platform supported in the WebGestaltR can be found by the function *listReferenceSet*.If *referenceGeneFile* and *refereneceGene* are emphNULL, WebGestaltR will use the *referenceSet* as the reference gene set. Otherwise, WebGestaltR will use the user uploaded reference set for the enrichment analysis.

15. *minNum*: WebGestaltR will exclude the categories with the number of annotated genes less than *minNum* for the enrichment analysis. The default is *10*.

16. *maxNum*: WebGestaltR will exclude the categories with the number of annotated genes larger than *maxNum* for the enrichment analysis. The default is *500*.

17. *fdrMethod*: For the ORA method, WebGestaltR supports five FDR methods: *holm*, *hochberg*, *hommel*, *bonferroni*, *BH* and *BY*. The default is *BH*.

18. *sigMethod*: Two significant methods are available in the WebGestaltR: *fdr* and *top*. *fdr* means the enriched categories are identified based on the FDR and *top* means all categories are ranked based on FDR and then selected top categories as the enriched categories. The default is *fdr*.

19. *fdrThr*: The significant level for the *fdr* method. The default is *0.05*.

20. *topThr*: The threshold for the *top* method. The default is *10*.

21. *dNum*: The number of enriched categories visualized in the DAG (directed acyclic graph) of the final report if the selected enrichment database contains a DAG structure. The default is *20* and the maximum is *100*. A larger *dNum* will increase the running time.

22. *perNum*: The number of permutations for the GSEA method. The default is *1000*.

23. *lNum*: The number of categories with the output leading edge genes for the GSEA method. The default is *20*. *Note*: GSEA first ranks the categories based on NES (normalized enrichment score) instead of FDR and then outputs the leading edge genes for top *lNum* categories. Because NES does not necessarily decrease with the increase of the FDR, using *sigMethod* defined in WebGestaltR to identify the significant categories may cause some categories with outputted leading edge genes are not included in the final result even if the number of significant categories is larger than *lNum*.

24. *is.output*: If *is.output* is TRUE, WebGestaltR will create a folder named by the *projectName* and save the mapping results, GO slim summary, enrichment results and an user-friendly HTML report in the folder. Otherwise, WebGestaltR will only return an R *data.frame* object containing the enrichment results. If hundreds of gene list need to be analyzed simultaneous, it is better to set *is.output* as FALSE.

25. *outputDirectory*: The output directory for the results.

26. *projectName*: The name of the project. If *projectName* is NULL, WebGestaltR will use time stamp as the project name.

27. *keepGSEAFolder*: If *keepGSEAFolder* is TRUE, WebGestaltR will keep all folders generated from GSEA tool that contain all figures and tables related to the GSEA analysis.

28. *hostName*: The server URL for accessing the data. User can use *listArchiveURL* function to get all archive version URL.

3.2 Output

The WebGestaltR function not only outputs the user-friendly HTML report containing the ID mapping table, GO Slim summary result and the enrichment analysis result but also outputs an R object containing the enrichment analysis result.

3.3 NOTE

Because WebGestaltR will read the data from the server for the ID mapping and GO Slim summary, the running time for the WebGestaltR function will be also based on the internet speed. Generally, it will take around one minute to perform the whole analysis. Because of the huge number of the Gene ontology categories, running GSEA analysis for these categories may take one to six minutes based on the different size of the uploaded gene rank list. Decreasing the parameter *maxNum* can reduce the running time.

If the user has any problem for the Network Visualization in the HTML report, please follow the instruction in the http://cytoscapeweb.cytoscape.org/tutorial.

4 Batch analysis of WebGestaltR

The WebGestaltR_batch function can perform the batch analysis for multiple gene or ranked gene lists.

4.1 Input

1. *interestGeneFolder*: The folder containing multiple interesting gene files. If *enrichMethod* is *ORA*, the extension of all files should be *txt* and each file can only contain one column: the interesting gene list. If *enrichMethod* is *GSEA*, the extension of each file should be *rnk* and the file should contain two columns separated by tab: the gene list and the corresponding scores.

2. *interestGeneType*: The ID type of the lists in all files. The supported ID type of the WebGestaltR_batch for the selected organism can be found by the function *listIDType*. If the *organism* is *others*, users do not need to set this parameter. NOTE: the ID type in all files should be the same.

3. *is_parallel*: If *is_parallel* is TRUE, WebGestaltR_batch will use parallel computing to simultaneously analyze the lists in all files.

4. *nThreads*: The number of cores used for parallel computing.

The description of other parameters can be found in the description of the *WebGestaltR*.

4.2 Output

If *is.output* is TRUE, each enriched result will be saved in a folder with the name containing the input file name under the *outputDirectory*. Otherwise, the WebGestaltR_batch function will return a list object containing all results.

If there are errors during the calculation, error message can also be found in the returned list object.

5 Get server URL

The listArchiveURL function can list the server URL of each version of the data update. Users can select different server URL as the *hostName* to perform the enrichment analysis.

```
> library("WebGestaltR")
> url <- listArchiveURL()
```

```
Version:Current Version      URL:http://www.webgestalt.org/
```

5.1 Output

The serve URL for all archive versions.

6 Check the format of the uploaded data

The formatCheck function can check the format of the gene list or ranked gene list file or object uploaded to the WebGestaltR for the analysis.

```
> library("WebGestaltR")
> geneFile<-system.file("extdata","interestingGenes.txt",package="WebGestaltR")
> interestGene <- formatCheck(dataType="list",inputGeneFile=geneFile,inputGene=NULL)
```

6.1 Input

This section describes the arguments of the *formatCheck* function: 1. *dataType*: Currently, this function supports 2 data type: ”list” means the uploaded file or data is a gene list and ”rnk” means the uploaded file or data is a ranked gene list with two columns (genes and scores).

2. *inputGeneFile*: The uploaded data file. If the dataType is ”list”, the file extension should be ”txt” and have only one column. If the dataType is ”rnk”, the file extension should be ”rnk” and have two columns (genes and scores).

3. *inputGene*: The uploaded R object. If the dataType is ”list”, the R object should be a vector. If the dataType is ”rnk”, the R object should be a data.frame and have two columns (genes and scores).

6.2 Output

The formatCheck function will return error if the data format is incorrect. Otherwise, it will return the processed data for the analysis.

7 ID Mapping

The IDMapping function can map one id type supported by the WebGestaltR to any other id type supported by the WebGestaltR. This function can perform the ID mapping for three types of data: a gene list, a gene list with the scores and a gmt file.

```
> library("WebGestaltR")
> interestGeneFile <- system.file("extdata","interestingGenes.txt",package="WebGestaltR")
> idmap <- IDMapping(organism="hsapiens", dataType="list", inputGeneFile=interestGeneFile, inputGene=
```

7.1 Input

This section describes the arguments of the *IDMapping* function: 1. *inputGeneFile*: three types of the files are supported for uploading the data: a *txt* file for a gene list, a *rnk* file for a gene list with scores (separated by tab) and a *gmt* file (first column is category id, second one is external link of the cagetory and other columns are the annotated genes. all columns are separated by tab).

2. *dataType*: The IDMapping function can perform the ID mapping for three types of data: *list* (a gene list), *rnk* (a ranked gene list) and *gmt* (a gmt file).

3. *inputGene*: two types of the R objects are supported for uploading the data: an R *vector* object for a gene list and an R *data.frame* object for a gene list with scores.

4. *sourceIdType*: the ID type of the uploaded data. The supported ID type of the WebGestaltR for the selected organism can be found by the function *listIDType*.

5. *targetIdType*: the target ID type for ID mapping. The supported ID type of the WebGestaltR for the selected organism can be found by the function *listIDType*.

6. *is_outputFile*: if is_outputFile is TRUE, the mapping results will be outputted to a file.

7. *outputFileName*: the output file name. No extension in the file name and the function will add the extension based on the input data type.

The description of other arguments can be found in the description of the *WebGestaltR* function.

7.2 Output

The IDMapping function will output an R data.frame obejct with three types of structure based on the three types of the input data. If the *targetIdType* is one of *entrezgene*, *genesymbol* and *genename*, the output object will contain four columns for a gene list (*userid*, *genesymbol*, *genename* and *entrezgene*), five columns for a gene list with scores (*userid*, *genesymbol*, *genename*, *entrezgene* and *score*) and six columns for a gmt file (*geneset*, *link*, *userid*, *genesymbol*, *genename* and *entrezgene*). If the *targetIdType* is other ID type, the data.frame object will add one more column *targetid*.

7.3 NOTE

Because the IDMapping function will read the mapping tables from the server, the running time for the WebGestaltR function will be also based on the internet speed. Generally, it will take around 20 seconds to perform the ID mapping.

8 GO slim summary

The *GOSlimSummary* function can summary the gene list based on the biological process, cellular component and molecular function ontologies of the GO Slim data sets. The summary result will be plotted as three bar plots and outputted to the PDF file.

```
> library("WebGestaltR")
> geneListFile <- system.file("extdata","GOSlimExample.txt",package="WebGestaltR")
> geneList <- read.table(geneListFile, header=FALSE, sep="\t", stringsAsFactors=FALSE)
> geneList <- as.vector(as.matrix(geneList))
> outputFile <- paste(getwd(),"/GOSlimSummary",sep="")
> GOSlimSummary(organism="hsapiens", genelist=geneList, outputFile=outputFile, outputType="pdf")

NULL
```

8.1 Input

This section describes the arguments of the *GOSlimSummary* function:

1. *genelist*: an R *vector* object containing a gene list. GOSlimSummary only supports NCBI EntrezGene ID for the summary. For other ID types, please first use *IDMapping* function to map to the EntrezGene ID.

2. *outputFile*: the output file name.

3. *outputType*: The output file extension that can be *pdf*, *png*, or *bmp*.

The description of other arguments can be found in the description of the *WebGestaltR* function.

8.2 Output

The *GOSlimSummary* function will output a pdf, png or bmp file with three bar plots.

8.3 NOTE

Because the GOSlimSummary function will read the GO Slim data from the server, the running time for the GOSlimSummary function will be also based on the internet speed. Generally, it will take around 20 seconds to perform the summary analysis.

9 Read GMT file

The *readGMT* function can read the GMT file and transform to an R matrix object.

9.1 Input

This section describes the argument of the *readGMT* function:

1. *gmtFile*: The GMT file with the extension *gmt*.

9.2 Output

An R matrix object containing three columns: category ID, external link of the category and the annotated genes.

10 List organisms

The *listOrganism* function can list all supported organisms in the WebGestaltR.

```
> library("WebGestaltR")
> organism <- listOrganism()
```

10.1 Output

All organisms supported in the WebGestaltR.

11 listGeneSet

The *listGeneSet* function can list all available gene sets for the selected organism in the WebGestaltR.

```
> library("WebGestaltR")
> geneSet <- listGeneSet(organism="hsapiens")
```

11.1 Input

This section describes the argument of the *listGeneSet* function:

1. *organism*: Currently, the *listGeneSet* function supports 12 organisms. Users can use the function *listOrganism* to check the available organisms.

11.2 Output

All functional categories supported by the WebGestaltR for the selected organism.

12 listIDType

The *listIDType* function can list all available ID types for the selected organism in the WebGestaltR.

```
> library("WebGestaltR")
> idType <- listIDType(organism="hsapiens")
```

12.1 Input

This section describes the argument of the *listIDType* function:

1. *organism*: Currently, the *listIDType* function supports 12 organisms. Users can use the function *listOrganism* to check the available organisms.

12.2 Output

All ID types supported by the WebGestaltR for the selected organism.

12.3 Output

All functional categories supported by the WebGestaltR for the selected organism.

13 listReferenceSet

The *listReferenceSet* function can list all existing reference sets for the selected organism in the WebGestaltR.

```
> library("WebGestaltR")
> referenceSet <- listReferenceSet(organism="hsapiens")
```

13.1 Input

This section describes the argument of the *listReferenceSet* function:

1. *organism*: Currently, the *listIDType* function supports 12 organisms. Users can use the function *listOrganism* to check the available organisms.

13.2 Output

All reference sets existing in the WebGestaltR for the selected organism.