

Proyecto Colaborativo de Desarrollo de Software

Nombre del proyecto: R-RegulonDB

Responsable principal: José Alquicira Hernández, Heladia Salgado

Breve planteamiento del problema:

RegulonDB es una base de datos que almacena información la regulación transcripcional de la bacteria *Escherichia coli* K12.

Investigadores de las áreas de system biology y de genomics acceden a RegulonDB para obtener información de su interés.

A pesar de que RegulonDB tiene opciones para usar los datos como son servicios web y archivos de textos plano previamente procesados, no cuenta con una herramienta en algún lenguaje de programación enfocada al análisis de datos, de tal forma que facilite tanto la extracción y el análisis de los datos de los objetos biológicos almacenados en RegulonDB.

Objetivo del proyecto:

1. Desarrollar un paquete en R que nos permita extraer, manipular y visualizar la información de RegulonDB.
2. Definir e implementar un conjunto de funciones que respondan a un conjunto de preguntas biológicas.

Datos con los que se cuenta:

1. Archivo SQLite con los datos de los objetos de RegulonDB. Cada tabla o MART del archivo SQLite representa un objeto, y contiene varias columnas que representan los atributos de los objetos. Algunas propiedades o atributos tienen más de un valor, en realidad son tablas de datos (Puedes visualizar las tablas usando la herramienta DB Browser for SQLite - <http://sqlitebrowser.org/>)

Objetos

- GENE
- NETWORK
- TF

Resultado ideal que debe generar el software:

1. Librería con las funciones creadas en R para hacer uso de los datos de RegulonDB. La librería debe poder instalarse de la siguiente forma:

```
% devtools::install_github("user/r-regulondb")
% library(' r-regulondb ')
```

2. Funciones disponibles en la libreria

a) listDatasets() lists the biological objects available in the database

```
> listDatasets()
[1] "GENE" "OPERON"
```

b) listAttributes() lists the columns of a particular biological dataset or mart.

```
> listAttributes("GENE")
[1] "gene_id" "gene_name"
[2] "gene_posleft"
```

c) getAttr() is the main function of the library. It allows users to filter and retrieve data from a particular biological object or mart.

getAttr() has five parameters:

Attributes: refers to the information to be retrieved.

Filters: species a restriction of the query.

Values: values associated to each filter.

mart: biological object or data mart

Usage example

```
> Gen_Exa<-getAtt(attributes = c("gene_id",
"gene_name", "gene_type","gene_strand" ),
+ filters = c("gene_strand", "gene_type"),
+ values = c("reverse", "Pseudo Gene"),
+ mart = "GENE",
+ cond = "AND")
> head(Gen_Exa)
```

	gene_id	gene_name	gene_type	gene_strand
1	ECK120000393	glpR	Pseudo Gene	reverse
2	ECK120000652	nmpC	Pseudo Gene	reverse
3	ECK120000854	rph	Pseudo Gene	reverse
4	ECK120001559	ychG_2	Pseudo Gene	reverse

Other useful functions

Gene

- The **getGeneRegulation** function allows the extraction of all known annotated regulation for a gene in RegulonDB.

```
getGeneRegulation(genes, format= c('multirow', 'onerow',  
'table'), output.type = c('TF','gene'))
```

example:

```
gene.regulation <- GetGeneRegulation(genes = c('araB',  
'araC', 'araE', 'araF'), format = 'multirow', type = 'TF')
```

Output

- ****multirow****. Raw data format. Each row represents a regulatory interaction

```
```\ngenes  regulators  effect  
araB AraC +-
araB CRP -
araC AraC +-
araC CRP +
```\n
```

- ****onerow****. One row per regulated gene

```
```\ngenes  regulators  
araB AraC(+), CRP(-)
araC araC(-), CRP(-)
araF araC(+), CRP(-)
araE araC(+), CRP(-)
```\n
```

- ****table****. rows are regulated genes (input) and columns regulators

```
          AraC      CRP  
araB      +-      -  
araC      +-      +  
araE      NA      NA  
araF      NA      NA
```

Transcription factor

- The **GetRegulatedGenesByTF** is the function that gets all genes regulated by a TF.

```
GetRegulatedGenesByTF(tf)
```

Example:

```
regulated_genes <- GetRegulatedGenesByTF(tf = 'AraC')
```

Output

araA
araB
araC
araD
araE
araF
araG
araH
araJ
xylA
xylB
ydeM
ydeN

Referencias Útiles:

1. <http://sqlitebrowser.org/>
- 2.