

# **Manual para uso de herramienta de Ciclos y su visualización en Cytoscape**

**Programa de Genómica Computacional, Centro de Ciencias Genómicas, UNAM  
Cuernavaca, Morelos**

María Georgette Femerling Romero

**Contacto:** [gfemer@lcg.unam.mx](mailto:gfemer@lcg.unam.mx) / [georgette.femerling@gmail.com](mailto:georgette.femerling@gmail.com)

**Supervisado por:** Julio Collado-Vides, Daniela Ledezma-Tejeida, Laura Gómez, Víctor Hugo  
Tierrafría, Claire Rioualen

## **Tabla de contenido**

1) Objetivo General.....	3
2) Dependencias del programa.....	3
3) Archivos de la librería.....	4
4) Instalación.....	4
5) Archivos de Input.....	4
6) Encontrar ciclos .....	4
7) Archivos de salida.....	5
8) Importar archivos a cytoscape para generar la red .....	6

## 1) Objetivo General

El algoritmo busca integrar información de regulación génica y de metabolismo con el objetivo de encontrar relaciones no obvias entre funciones a partir de conexiones metabólicas y regulatorias. La herramienta busca también comparar los ciclos que se encuentran con los datos de RegulonDB con los ciclos que se encuentran con una red de regulación de otra fuente. Funciona mediante la búsqueda de “ciclos” partiendo de factor de transcripción inicial y recuperando las categorías funcionales de MultiFun y los genes regulados por el TF inicial que forman parte también de cada categoría funcional. A partir de esto, se recuperan genes en categorías funcionales distintas que catabolizan reacciones con metabolitos en común. El ciclo se cierra cuando se encuentra una conexión directa (el gen encontrado es regulado por el TF inicial) o indirecta (el gen encontrado es regulado por otro TF regulado por el TF inicial) con el TF que se inició el ciclo. La esquematización gráfica se encuentra en la figura 1.

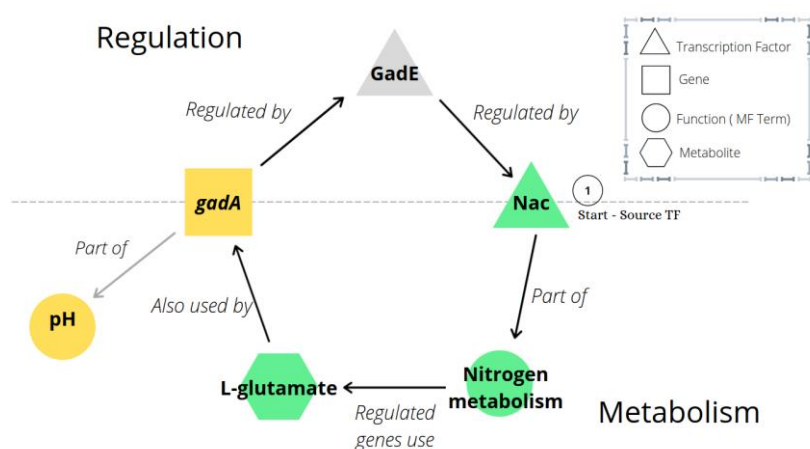


Figura 1. Esquematización de cómo funciona el algoritmo.

Una vez que se encuentra el ciclo se toman las categorías funcionales del gen encontrado y se establece una conexión con la categoría funcional inicial como se muestra en la figura 2.

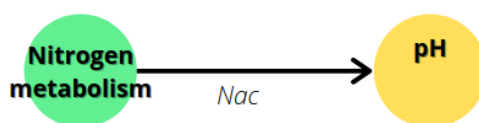


Figura 2. Conexión entre categorías funcionales encontrada a partir del ejemplo de la fig 1.

El pipeline encuentra estos ciclos para la red tf-gen de RegulonDB y para una red distinta y los compara, generando como output una red de funciones metabólicas con anotaciones sobre el estatus de la interacción (conocida o no) dependiendo de si es una interacción que ocurre en la red de regulación de RegulonDB o no.

## 2) Dependencias del programa

- Interpreter de Bash
- Pathway tools API
- perl 5 (se corrió con perl v5.16.3)

- módulos: perlcyc, strict, warnings
- python 2.7
  - librerías: collections, datetime, sys, os, re

### 3) Archivos de la librería

Los archivos de la librería se obtienen de Ecocyc / pathway tools

- *gene\_GenProtEc.txt* - contiene las anotaciones de Multifun de cada gen
- *gene\_links.txt* – contiene nombres, sinónimos e identificadores de los genes, se encuentra como *gene\_links.dat* en los archivos de instalación de pathway tools.

### 4) Instalación

----- en construcción -----

### 5) Archivos de Input

1. Red formato tf-gen descargada de RegulonDB, archivo separado por tabs

```
#
#
# Columns:
# (1) Transcription Factor (TF) name
# (2) Gene regulated by the TF (regulated gene)
# (3) Regulatory effect of the TF on the regulated gene (+ activator, - repressor, +- dual, ? unknown)
# (4) Evidence that supports the existence of the regulatory interaction
# (5) Evidence Type [Weak, Strong, Confirmed] For details see: http://regulondb.ccg.unam.mx/evidenceclassification
#
AccB accB repressor [GEA, IMP] Weak
AccB accC repressor [GEA, IMP] Weak
AcrR acrA repressor [APIORCISFBSCS, BCE, BPP, GEA] Strong
AcrR acrB repressor [APIORCISFBSCS, BCE, BPP, GEA] Strong
AcrR acrR repressor [AIBSCS, APIORCISFBSCS, BCE, BPP, GEA] Strong
AcrR flhC repressor [APIORCISFBSCS, GEA] Weak
AcrR flhD repressor [APIORCISFBSCS, GEA] Weak
```

2. Red nueva formato tf-gen, archivo separado por tabs

```
Ada csrB
Ada truC
Ada yqcC
AdiY dhaK
AdiY dhaL
AdiY dhaM
AdiY dhaR
AdiY glnA
AdiY glnG
AdiY glnL
```

3. Lista de los TFs de inicio, un TF por línea

```
Ada
AdiY
AgaR
AllR
ArcA
ArgP
ArgR
ArsR
```

4. Directorio de la librería

### 6) Encontrar ciclos

El pipeline para correr ciclos se controla con un solo script con el cual calcula los ciclos para ambas redes (de regulon y la “nueva”) y genera distintas tablas y reportes intermedios, así como las redes de funciones finales.

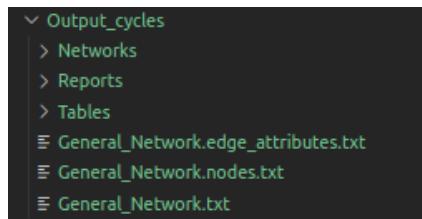
Correr como:

```
$ bash bin/cycles_master.sh [Lista de Tfs] [Red Nueva] [Red RegulonDB] library/ [directorio de salida]
```

Ejemplo:

```
[gfemer@sinik ~]$ bash bin/cycles_master.sh Input_tf_lists/filtered_ecoli_potint_20201004_input-tfs.txt ../GUs/Networks/filtered_network_from_potential_interactions_FC1_up.txt ../GUs/Networks/Regulon10.7_tf_gene.txt library/ Prueba_output
```

## 7) Archivos de salida



El pipeline produce como output 3 directorios y 3 archivos a parte.

- *Networks*: Contiene un archivo de red de Multifuns para cada TF analizado.

La red tiene un formato MF source, MF target, Status of interaction (Known or New)

```
BC-1.8.3:nitrogen metabolism(35)- BC-1.1.1:carbon compounds(270)- Known
BC-1.8.3:nitrogen metabolism(35)- BC-1.1.3:amino acids(55)- Known
BC-1.8.3:nitrogen metabolism(35)- BC-1.3.4:TCA cycle(18)- New
BC-1.8.3:nitrogen metabolism(35)- BC-1.3.5:fermentation(31)- New
BC-1.8.3:nitrogen metabolism(35)- BC-1.5.1.17:alanine(7)- New
```

- *Reports*: Contiene dos archivos por cada TF analizado, uno para lo encontrado en RegulonDB y otro para lo encontrado en la red que se compara. Los reportes consisten en los ciclos encontrados por cada MultiFun del TF, así como la interacciones regulatorias y los metabolitos que conforman el ciclo.

```
# From bin/get_cycles.v6.pl on Wed Dec 23 17:53:03 2020
# Input Network= Output_cycles/tmp/Regulon_general_regulons.txt
# Genes marked with * participate in a reaction with a DeltaG less than -58.5
# MF terms not considered to find cycles: Represor, Transcriptional level, regulon, global, operon, Transcription related, activator.
# TF name: PurR
# Gene ID: EG10800
# Multifun Terms associated with PurR: BC-1.5.2.2:pyrimidine biosynthesis(12), BC-1.5.1.16:histidine(10), BC-1.5.2.1:purine biosynthesis(15)
# Multifun ID: BC-1.5.2.2
# Multifun: pyrimidine biosynthesis(12)
# Genes regulated by PurR that are also part of the Multifun: carA,carB,pyrC,pyrD
# Compounds of PurR: a menaquinone,hydrogencarbonate,ammonium,(<i>S</i>)-dihydroorotate,carbamoyl phosphate,a menaquinol,an electron-tr
# Interactions- Associated Multifun Shared Metabolite
PurR- gcvT- BC-1.7.17:formyl-THF biosynthesis(15)- ammonium
PurR- gcvT- BC-1.1.3:amino acids(55)- ammonium
PurR- gcvP- BC-1.7.17:formyl-THF biosynthesis(15)- ammonium
PurR- gcvP- BC-1.1.3:amino acids(55)- ammonium
PurR- codA- BC-1.7.33:nucleotide and nucleoside conversions(56)- ammonium
PurR- purK- BC-1.5.2.1:purine biosynthesis(15)- hydrogencarbonate
```

- *Tables*: Contiene un archivo por cada TF analizado. Integra la información que se encuentra en los reportes y la compara. Divide las interacciones regulatorias en directas e indirectas y en conocidas y no conocidas. Contiene también los metabolitos que unen a las funciones.

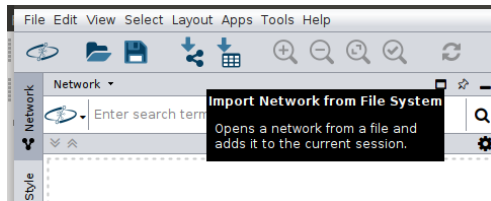
```
# From get_cycles_output2.py
# Date: 2020-12-23
# Genes marked with * participate in a reaction with a DeltaG less than -58.5
# Multifun Terms marked with * are new relative to RegulonDB network
#
# Input TF: Nac
# Multifun Terms associated with Nac: BC-2.2.2:Transcription related(286),BC-3.3.1:operon(120),BC-3.1.2.3:repressor(154),BC-1.8.3:nitrogen metabolism(35)
# Start MF terms: BC-1.8.3:nitrogen metabolism(35)
#
Start MF(genes in MF)- Multifun Term(genes in MF)- Direct Regulation known Direct Regulation unknown Indirect Regulation(TF/gene) known
BC-1.8.3:nitrogen metabolism(35)- BC-1.1.1:carbon compounds(270)- gabP - - - gabP/5-aminopentanoate,gabP/4-aminobutanoate-
BC-1.8.3:nitrogen metabolism(35)- BC-1.1.3:amino acids(55)- metB,iaaA,dadA- FeaR/tyrA- dadA/ammonium,metB/ammonium,iaaA/ammonium
BC-1.8.3:nitrogen metabolism(35)- *BC-1.3.4:TCA cycle(18)- ppc- - - ppc/hydrogencarbonate- hydrogencarbonate
BC-1.8.3:nitrogen metabolism(35)- *BC-1.3.5:fermentation(31)- ppc- - - ppc/hydrogencarbonate- hydrogencarbonate
BC-1.8.3:nitrogen metabolism(35)- *BC-1.5.1.17:alanine(7)- - - YqeI/alaA- YqeI/alaA/L-glutamate- L-glutamate
```

- *General\_Network.nodes.txt* – Archivo para importar a cytoscape como red. Formato MF source – Mf target, separado por tabs.
- *General\_Network.edge\_attributes.txt* - Archivo para importar a cytoscape como atributos de las aristas de la red que se creó con el archivo anterior. Separado por tabs.

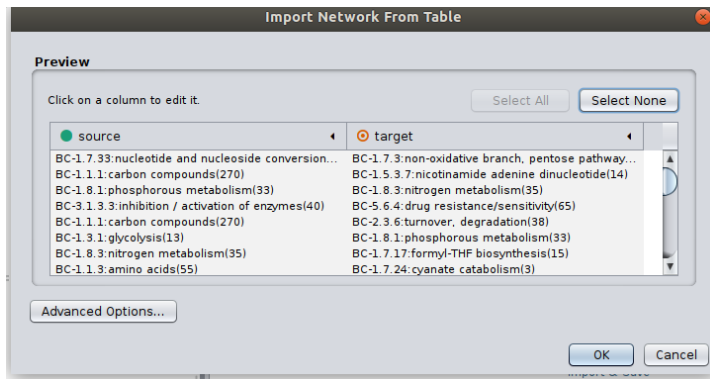
- *General\_Network.txt* – Archivo que integra todos los datos de los archivos de cytoscape en una sola tabla. Estos archivos juntan todas las interacciones que se encontraron por TF para crear una red que contenga toda la información de los TFs de la lista de entrada.

## 8) Importar archivos a cytoscape para generar la red

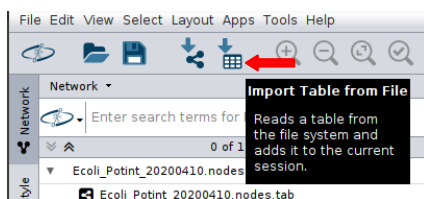
Los archivos necesarios para generar la red general en cytoscape son *General\_Network.edge\_attributes.txt* y *General\_Network.nodes.txt*.



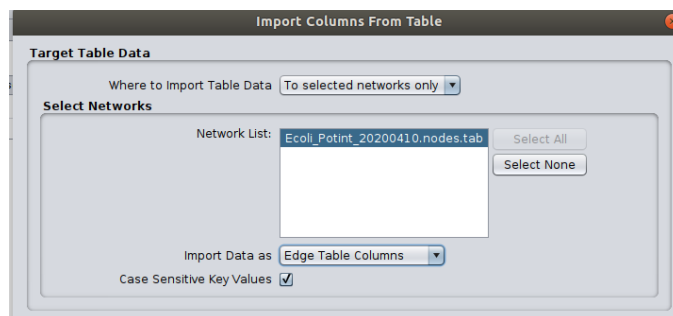
1. Abrir Cytoscape y seleccionar importar red de archivo. Importar *General\_Network.nodes.txt*



2. Seleccionar la primer columna como “source” y la segunda como “target” y darle “ok.”
3. Importar los atributos de las aristas seleccionando importar tabla de archivo. Importar *General\_Network.edge\_attributes.txt*

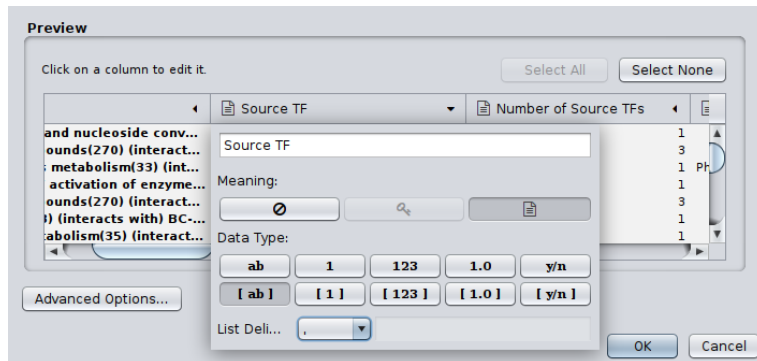


4. Seleccionar “To selected networks only” en la sección de Target Table Data y seleccionar la red anteriormente creada. Seleccionar “Edge table columns” en Import Data as.



5. Para el key, automaticamente se selecciona la columna de shared name. Para las otras columnas se tiene que editar el tipo de dato de la siguiente forma, al terminar darle click a “ok”:

- Source TF – lista delimitada por comas
- Number of Source Tf – Integer (automático)
- Known direct regulation – lista delimitada por comas
- Unknown direct regulation – lista delimitada por comas
- Known indirect regulation – lista delimitada por comas
- Unknown indirect regulation – lista delimitada por comas
- Direct Reg metabolite – lista delimitada por comas
- Indirect Reg metabolite – lista delimitada por comas
- All metabolites – lista delimitada por comas
- Number of Metabolites – Integer (automático)
- Type of interaction – String (automático)
- Status of interaction – String (automático)



6. Correr el analizador de redes como red no dirigida en Tools > Analyze Network > Ok.

\* Opcional para darle formato a la red

7. Ir a Style > Node

- Fill Color > Map, Column: Clustering Coefficient, Mapping Type: Continuous
- Shape > Default Value: Ellipse
- Size > Map, Column: Degree, Mapping Type: Continuous, Current Mapping: Node size min: 30, Max: 120 (ajustar dependiendo de la red).
- Label Font Size > Default Value: 20
- Check Lock node with and height

8. Ir a Style > Edge

- Stroke Color > Map, Column: Status of interaction, Mapping Type: Discrete Mapping, Elegir un color para cada tipo (In RegulonDB – Azul #0000CC, Known – Morado #660066, New – Rojo #FF0000)
- Width > Map, Column: Number of Source TFs, Mapping Type: Continuous, Current Mapping: Edge Width min: 2.0, max: 15 (ajustar dependiendo de la red)

9. Guardar como pagina web en File > Export > Network to Web Page. Seleccionar Export as: Full Web Application. Guardar en la ubicación deseada y darle “Ok.” Para abrir la página web hay que descomprimir el directorio y abrir index.html.

10. Guardar tambien como sesión de cytoscape en File > Save As. Guardar en ubicación deseada y darle “Ok.”