**rg3m**

**Adam Podgorny & Cory Jenkinson**

**Care of: Oakley Lab, University of Kansas**

**Overview**

This script imports a resistance gene (query) csv file and a secondary metabolite gene (target) csv file containing the genomic coordinates of the genes and then returns matches/hits (potential BGCs with resistance genes) within a user-specified genomic distance cutoff.

**Algorithm**

The input consists of a resistance gene (query) csv file, secondary metabolite gene (target) csv file, a user-specified cutoff distance in base pairs, and a few optional settings. The optional 'nocheck' flag affects filtering behavior for the query file, removing the requirement for each genome to contain at least two query genes (a housekeeping and putative resistance gene). Another optional setting allows the user to control the maximum gene size whereby the genes above the threshold are excluded; the default maximum gene size is 50,000 base pairs. The third optional setting is a homolog search mode that finds resistance gene homologs in genomes that contain hits of potential BGCs with resistance genes.

The resistance gene file is read in, such that a table is created. Each entry contains the organism name, scaffold/chromosome, gene start and end locations, E-value, and % identity. The center of the gene is calculated by the simple geometric average of the start and end locations. If the optional 'nocheck' parameter is not used, genomes that do not contain at least two query genes are filtered out. The secondary metabolite file is then read in to create a gene table in this same manner but without any filtering by redundancy.

Each entry in the resistance gene table is then tested against each entry in the secondary metabolite gene table. If the organism names match, the entries are checked for a scaffold/chromosome match. If the scaffolds/chromosomes match, genomic overlap and inclusion within the user-specified distance cutoff are then assessed. If the putative hit is within the user-specified distance cutoff and the two genes do not overlap, the hit is accepted and output.

The accepted hits are then printed to screen. If an output file is specified, these entries are output in csv format. Each hit entry contains the ‘Organism Name’, ‘Resistance Gene Scaffold’, ‘Resistance Gene Start’, ‘Resistance Gene End’, ‘Resistance Gene E-value’, ‘Resistance Gene % Identity’, ‘SM Gene Scaffold’, ‘SM Gene Start’, ‘SM Gene End’, and the ‘Distance’ between the center of the resistance and SM genes.

If the homolog mode is activated, resistance genes from the hits will be used to rescan the resistance gene file for all homologs in the same organism. The homologs will then be output to the specified file name.

**Usage**

python rg3m --cutoff <cutoff in bp> --resistance\_gene <resistance gene csv file> --sm\_gene <SM gene csv file> [--out outputs hits as csv file] [--rghomologs resistance gene homolog mode and outputs homologs to csv file] [--nocheck] [--gene\_length\_cutoff <cutoff in bp>]

The Python version is up to the user’s choice, although this was developed on a Python2 base. It requires no libraries beyond the libraries that come standard with Python2.x and 3.x to make deployment simpler.

* --cutoff - This parameter controls the distance from the center of the query gene to the center of the target gene and determines which hits are accepted or rejected. The units are in base pairs.
* --resistance\_gene - The csv file with the resistance genes.
* --sm\_gene - The csv file with the secondary metabolite genes.
* --out - An optional parameter which allows the user to dump the results to the specified csv file.
* --nocheck - An optional parameter which waives the requirement for each genome (from the resistance gene csv file) to contain at least two query genes.
* --rghomologs - An optional parameter which turns on the homolog mode for the resistance genes and then outputs them to the specified csv file.
* --gene\_length\_cutoff - An optional parameter which controls the maximum gene size. Genes above the threshold are excluded. The default maximum gene size is 50,000 base pairs.

**For New Python and MycoCosm Users:**

To create csv files from the MycoCosm homepage, users can click on the fungal group of interest, for example ‘Eurotiomycetes’, ‘Ascomycota’, ‘Fungi’, etc., and then from the drop-down menu they can select the ‘Search’ or ‘BLAST’ options.

When using the ‘Search’ function, a variety of search options are available, and we have used the ‘Keywords’ and ‘KOG terms’ (euKaryotic Orthologous Groups) search options to generate csv files.

When using the BLAST function (we like to use the TBLASTN option on the BLAST page), on the results page, after clicking the ‘Configure This Screen’ button, from the drop-down menu the user should type in the number of hits found into the ‘Showing Top \_\_\_ Hits Per Group By:’ box, click ‘Submit’, and then click the ‘Alignment Hit Table’ box; after this, the user can export the hits as a csv file using the ‘Hits Excel Spreadsheet’ option from the drop-down menu and clicking the export button.

We downloaded our csv files from MycoCosm. Some csv files downloaded from MycoCosm will contain a column entitled ‘Location’ which needs to be deleted for rg3m to function.

Many times, csv files downloaded from MycoCosm will contain commas. These commas need to be deleted for rg3m to function. We suggest using “find and replace all” function in Numbers or Microsoft Excel to delete these commas.

If csv files are modified in programs such as Numbers or Microsoft Excel, users need to remember to save/export the modified files as csv files; rg3m only works with csv files.

For rg3m to function, the csv files for both the resistance genes and SM genes require columns for the: ‘Organism Name’, ‘Scaffold’, gene ‘Start’, and gene ‘End’. The columns for ‘% Hit Identity’ and ‘EValue’ are not required in the resistance gene or SM gene csv files for rg3m to function. The csv files for both the resistance genes and SM genes can have additional columns.

For rg3m to function the titles of the columns must be as follows:

The column for ‘Organism Name’ can also be entitled ‘Organism’ or ‘organism’.

The column for ‘Scaffold’ can also be entitled ‘scaffold’, ‘Chromosome’, ‘chromosome’, or ‘Hit Name’.

The column for the gene ‘Start’ can also be entitled ‘Hit Start’, ‘hit start’, or ‘hit\_start’.

The column for gene ‘End’ can actually be named anything, but must be the column directly after the gene ‘Start’ column.

The column for ‘% Hit Identity’ can also be entitled ‘% identity’, ‘Percent Identity’, ‘Identity Percent’, ‘Per. Identity’, ‘Per Identity’, ‘Identity’, or ‘Identities’.

The column for ‘EValue’ can also be entitled ‘evalue’ or ‘Evalue’.

We suggest rg3m users to place the rg3m.py file, the resistance gene csv file, and the SM gene csv file on their desktop. On Macs in the ‘Terminal’ command-line interface, users can direct ‘Terminal’ to these three files by typing ‘cd Desktop’ and then pressing the ‘return’ button.

The following is an example of how to use rg3m after putting the necessary files on the desktop and writing ‘cd Desktop’ followed by pressing the ‘return’ button. If the user has the following files: ‘rg3m.py’, the resistance gene file ‘AN5784.csv’, and the SM gene file ‘NRPS.csv’ and wants to use a 70,000 bp distance cutoff between the potential resistance gene and SM gene, the following would be typed into ‘Terminal’ followed by pressing the ‘return’ button:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

If the user wants ‘Terminal’ to output the results as a csv file with the title ‘AN5784\_NRPS\_70kb.csv’ then the user would type the following into ‘Terminal’:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

--out AN5784\_NRPS\_70kb.csv

If the user also wants ‘Terminal’ to use ‘homolog’ mode and output the results as a csv file with the title ‘AN5784\_NRPS\_70kb\_homologs.csv’ then the user would type the following into ‘Terminal’:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

--out AN5784\_NRPS\_70kb.csv --rghomologs AN5784\_NRPS\_70kb\_homologs.csv

If the user also wants ‘Terminal’ to use the ‘nocheck’ function then the user would type the following into ‘Terminal’:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

--out AN5784\_NRPS\_70kb.csv --rghomologs AN5784\_NRPS\_70kb\_homologs.csv --nocheck

If the user also wants ‘Terminal’ to change the gene length cutoff from 50,000 bp to 60,000 bp then the user would type the following into ‘Terminal’:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

--out AN5784\_NRPS\_70kb.csv --rghomologs AN5784\_NRPS\_70kb\_homologs.csv --nocheck

--gene\_length\_cutoff 60000

Some computers might be running python3. To run rg3m on computers with python3, users should type the following:

python3 rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

rg3m can process the above commands in any order the user writes them in ‘Terminal’. For example, rg3m will accept any of the following orders of these commands and return the same result:

python rg3m.py --cutoff 70000 --resistance\_gene AN5784.csv --sm\_gene NRPS.csv

python rg3m.py --resistance\_gene AN5784.csv --sm\_gene NRPS.csv --cutoff 70000

python rg3m.py --sm\_gene NRPS.csv --resistance\_gene AN5784.csv --cutoff 70000

python rg3m.py --sm\_gene NRPS.csv --cutoff 70000 --resistance\_gene AN5784.csv

python rg3m.py --resistance\_gene AN5784.csv --cutoff 70000 --sm\_gene NRPS.csv