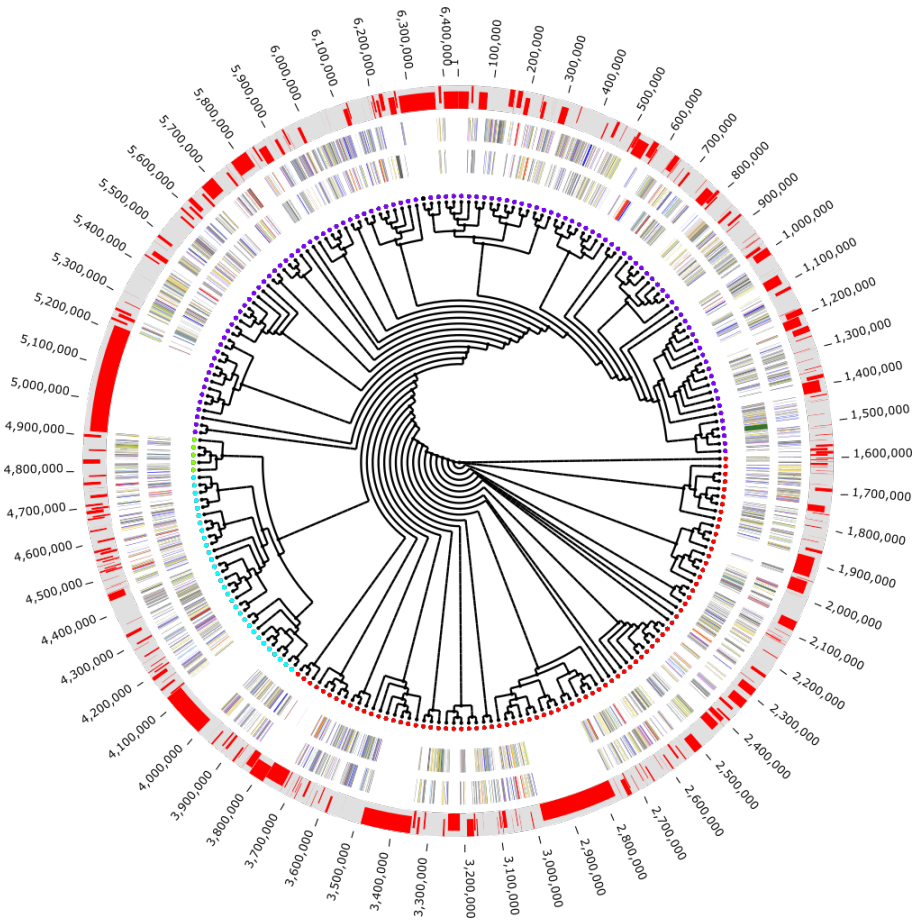
PanACEA (Pan-Genome Atlas and Chromosome Explorer and Analyzer) User Guide

Thomas H. Clarke

August 30, 2017



Contents

1 Introduction

The PanACEA software creates locally browsable pages to enable exploration of pan-chromosomes through interactive images at varying levels of detail, from a chromosome level view to multi-gene regions to single gene view. PanACEA uses a Perl script to dynamically generate the pages locally and provides the link to the initial main page for the beginner to start to use. It is designed to work with the PanOCT output, but can be altered to work with any set of pan-chromosome outputs using the file types shown in X. PanACEA is also able to incorporate multiple forms of gene annotation. Examples of GO term annotation, Anti-microbial resistance and Plasmid genes are shown in the manual. Finally, PanACEA can export publication quality graphics in multiple formats of multiple sections and levels, depending on the interest of the user.

2 Installation

To run PanACEA only requires a single Perl script (make_panacea.pl) and a panacea flatfile, shown later. However, additional files like config files can be used and the PanACEA flatfile can be generated from PanOCT output with a provided script. On our download site, we have included: the Perl scripts, several example config files, and this manual. All the necessary files, including the HTML, SVG and JavaScript files, are created by the Perl script. To download this and a version of the manual, please follow this link. PanACEA was optimized with the PanOCT result files, so we suggest using these results.

3 Generating PanACEA Files

The PanACEA Perl script is operated from the command line. There are several command option available while running the script which are listed below:

- **-i** input PanACEA Flatfile (described below). Required
- **-o** output directory where the html, svg, json, and JavaScript files used to run PanACEA. Default is current directory.
- **-n** Name used with the output. Default is "PanACEA".
- **-d** Web browers directory to access the output directory. Default is "file://" + Output directory
- **-f** config files that describes the files to use for the annotation of the genes. Default is no file
- **-t** newick files that describes the phylogeny of the genomes. Default is no tree file
- **-m** text files that describes the annotation of the genes. Only used if tree file is also included. Default is no file

- **-g** config files that describes the graphics and sizes. Default variables are listed below
- **-a** Directory containing the multiple alignments of the clusters. Default is to use the unaligned fasta
- **-h** Show the help page then quits

Example command line:

```
make_panacea_flatfiles.pl -d <PANGENOME_ROOT_OUTPUT> -o
<FLAT_FILE_NAME> -t Pangenome
make_panacea.pl -i <FLAT_FILE_NAME> -o ./test/ -n Test -f exam-
ples/func_file.conf -g examples/graphics.conf -t examples/outtree -m
examples/ABCD.grouping
```

3.1 PanACEA Input FlatFile

The PanACEA script requires a specialized flat file to create the images. An additional script is available to transcribe information from Gene Order programs like PanGenome and Rorey into the flat file. The flat file is in a modular format with the different levels of the image represented. START and END are required to be in all caps, as is the TYPE. The columns are all required to be tab seperated. The basic format of the modules is shown below:

```
START TYPE ID
Variable Value
Variable Value
....
END
```

The different types accepted in the file are:

- **CHROM** Chromosomes
- **CORE** Core genes and FGIs
- **FGR** Flexible Genomic Regions
- **CLUSTERS** Orthologous Gene Clusters

Each are described below along with the variables and possible values associated with each.

3.1.1 CHROM

PanACEA Pan-Chromosomes type describe large stretchs of continuous DNA. Expected variables include:

- **type** the type of the chromosome. Acceptible values are "cycle" (circular) or "chain" (linear). Required

- **sz** the size of the chromsome in basepairs. Required to be an integer. Required
- **is_core** whether the chromosome contains core genes, with 1 = yes and 0 = no. Required

An example module is shown below:

```
START CHROM 1
Variable Value
Variable Value
....
END
```

3.1.2 CORE

The Core type describes a section of Pan-Chromsome, either a gene cluster or an fGI (Flexible Genomic Island). The ID is expected to be unique for all the core genes. Expected variables include:

- **chr** the ID of the chromosome which the Core section is located.
- **start** the bp location on the chromosome where the Core section starts. For clusters, this is the 5' end. Required to be an Integer.
- **end** the bp location on the chromosome where the Core section ends. For clusters, this is the 3' end. Required to be an Integer.
- **type** the type of the core section. Accepted values are "fGI" or "CL". All others are ignored.
- **def** The gene name associated with the cluster. Ignored for fGIs

An example module is shown below:

```
START CHROM 1
Variable Value
Variable Value
....
END
```

3.1.3 FGR

The FGR type (Flexible Genomic Region) describes a section of the pan-chromosome that varies between genomes. Multiple fGRs are expected for each fGI. Expected varibles include:

- **order** the order of the genes associated with the FGR including the core genes when availble. The genes are colon separated. This is the ID of the fGR
- **cnt** the number of genomes that contains this FGR Required to be an Integer. Probably can just get from gen

- **st** the core gene on the 5' end of the fGR.
- **end** the core gene on the 3' end of the fGR.
- **gen** The genomes that contain the FGR in a colon separated list.
- **arr** The non-core gene clusters in the fGR. Probably just can get from order.

An example module is shown below:

```
START CHROM 1
Variable Value
Variable Value
....
END
```

3.1.4 Gene Clusters

The orthologous gene clusters describe the unique position in each

- **type** the order of the genes associated with the FGR including the core genes when availble. The genes are colon separated. This is the ID of the fGR
- **sz** the number of genomes that contains this FGR Required to be an Integer. Probably can just get from gen
- **st** the core gene on the 5' end of the fGR.
- **end** the core gene on the 3' end of the fGR.
- **gen** The genomes that contain the FGR in a colon separated list.
- **arr** The non-core gene clusters in the fGR. Probably just can get from order.

An example module is shown below:

```
START CHROM 1
Variable Value
Variable Value
....
END
```

3.2 PanOCT files

The PanACEA script uses several PanOCT files to create the Flatfile with make_panacea_flatfile.pl. A list of the files along with a description is shown first, with longer descriptions following which can be obtained by clicking on the file name. For ease, we use several abbreviations in this description, include fGI for flexible genome islands, fGR for flexible genomic regions.

- **Core.attfGI** The Core and Flexible Genomic Island attributes files gives the map location of each core gene and FGI on the core pan-chromosomes
- **shared_clusters.txt** A tab-separated table that lists the genes associated in each cluster
- **consensus.txt** A list giving core consensus list of the clusters associated with each assembly core and fGI
- **FGI_inserts.details** gives the core gene boundaries and the cluster lists for each fGI
- **fGI.att** a list giving the size of the different fGRs
- **combined.att** a list giving the location of each gene on the assembled contigs
- **frameshifts.txt** used to connect renamed genes to their original gene
- **singletons_clusters.txt** gets the cluster id of cluster with only a single gene

Given the input PanOCT directory, the script expects a structure like:

```
Input PanOCT Directory
├── combined.att
├── fastas
└── results
    ├── frameshifts.txt
    ├── shared_cluster.txt
    ├── singletons_clusters.txt
    └── fGI
        ├── Core.attFGI
        ├── consensus.txt
        ├── fGI.att
        └── FGI_inserts.details
```

3.2.1 Core.attFGI

Core.attFGI is located in the results/fGIs/ directory. The file is a tab separated file with columns containing:

1. the region (numbered)
2. the core cluster ID or the insertion region
3. start in bp
4. end in bp
5. name

6. type as either fGI or CL
7. length in bp

An example is below, with an insertion region shown in the first line and core genes shown the other lines.

```
1 CL_INS_1 1 29886 fGI_25;6 fGI 9962
1 CL_12 29887 30759 gluconolactonase CL 873
1 CL_13 31920 30760 P-protein CL 1161
1 CL_14 32262 31921 translation inhibitor protein RaiA CL 342
1 CL_15 33000 32263 outer membrane biogenesis protein BamD CL 738
1 CL_16 33001 33981 23S rRNA pseudouridine synthase D CL 981
1 CL_17 33982 34713 membrane protein CL 732
```

3.2.2 shared_cluster.txt

Shared_cluster.txt is located in the results directory. The file is a tab-separated table starting with a header line listing:

1 Cluster ID (Integer)

2 Number of Genes in the Cluster

3 String with the protein name

4 Genome containing the gene designated as the centroid

5 String with the gene name of the centroid

6 String with role id (not used)

7 String with attributes (not used)

8 ... N List of Genomes

The rest of the file consists of all clusters with the lines filled in. Missing data are shown as empty/skipped columns. Example data showing only a few genomes is shown below:

3.2.3 consensus.txt

Consensus.txt is located in the results/fGIs/ directory. The consensus file has a header line for each region starting with a # and has columns with:

1. region type (Assembly_Core or Assembly_fGI)
2. region id (integer)
3. region structure type (cycle or linear)
4. start in bp/aa
5. end in bp/aa

Example data showing the beginning of a core region in line one along with example genes below. Also, circular and linear fGRs are shown below.

```
#Assembly_Core 1 cycle 3886 3788113 aa
CL_12 + P 219 gluconolactonase
edge:218
CL_13 - P 219 P-protein
edge:218
CL_14 - P 219 translation inhibitor protein RaiA
```

```
#Assembly_fGI 6 cycle 5 2823 aa
CL_17600 - S 1 hypothetical protein
edge:1
```

```
#Assembly_fGI 16 chain 83 55575 aa
CL_1338 + S 141 cor protein
edge:80
```

3.2.4 FGI_inserts.details

FGI_inserts.details is located in the "results/fGIs/" directory. The fGI_inserts.details file is modular with a section for each fGI or insert. The header line for each module consists of the: region_ID *BoundingCoreGene*1_Direction *BoundingCoreGene*2_Direction

The rest of the module consists of list of flexible genome regions. While the Pangenome has several output options for the details files, all that is required is the first column which is a long string containing the ids and direction of all the genes in a given fGI arrangement. The string starts the start core gene, continues to list all the genes in the fGR, then shows the end core gene, such as `START_CORE[ID1][+-]:[ID2][+-]:[ID3][+-]..[IDn][+-]:STOP_CORE[IDx][+-]` where the `[+-]` shows the direction of the gene. If the START or STOP gene is unknown, this space is left blank. An example of both the header and the body is shown below:

```
CL_INS_1 12_5 4651_3
START_CORE12-:18890-:18891-:9325-:4650+:STOP_CORE4651-
START_CORE12-:18890-:18891-:20670+
START_CORE12-:STOP_CORE4651- 128
```

3.2.5 fGI.att

fGI.att is an attributes file similar to Core.att,with the columns:

1. the region (numbered)
2. a bounding gene (shown with CONTEXT) before or a member of the fGR
3. start in bp
4. end in bp
5. name
6. type as either fGI or CL
7. length in bp

An example is below including bounding genes in the first line and last line:

```
3 CONTEXT1:CL_18154 1894 2298 hypothetical protein CL 405
3 CL_14383 2299 2826 hypothetical protein CL 528
3 CL_14382 3042 2827 membrane protein CL 216
3 CL_14381 3294 3043 hypothetical protein CL 252
```

3.2.6 combined.att

Combined.att is an attributes file that containes the gene information:

1. the fasta gene ID
2. the cluster gene ID
3. start in bp
4. end in bp
5. gene name
6. genome ID

An example is below, demonstrating that 3`-5`sequences have a lower bp value in the end bp column (line 1,2,3) and that the list does not need to be in order.

```
NZ_AMGJ01000001 A965_RS0100015 1211 795 hypothetical protein 08XA1
NZ_AMGJ01000002 A965_RS0100470 103109 101724 nitrate/nitrite transporter NarK 08XA1
NZ_AMGJ01000002 A965_RS0100475 103895 103320 hypothetical protein 08XA1
```

3.2.7 frameshifts.txt

The Frameshifts.txt file is a corrective file that the PanGenome makes to rename certain genes that have been mistranslated. If this is not an issue with the Pangenomic software that is being used, it can be ignored. The file contains a tab-delineated table with the old name (the gene name in the combined.fasta file) in the first column and all subsequent columns are the new names.

```
>genome 08XA1
>asmbl_id NZ_AMGJ01000001
A965_RS01000000123715 A965_RS0100915
>asmbl_id NZ_AMGJ01000002
A965_RS01000000123720 A965_RS01000000123930
>asmbl_id NZ_AMGJ01000004
A965_RS01000000123790 A965_RS01000000124350
```

3.2.8 combined.fasta

Combined.fasta is a multi-fasta file that contains all the sequences in fasta format, with the fasta header ID equal to the gene name used in sharedCluster.txt. Even if the multiple sequence alignment has been turned on, the Combined.fasta is still used to check all the files.

3.2.9 singletons_clusters.txt

The Singleton_clusters.txt contains a list of all the clusters available only in a single genome. Since these are not in the Shared Clusters file above, it is required to connect the cluster id to the correct sequence.

1. the cluster gene ID number
2. number of members (should be 1)
3. gene name
4. role ids (should be 0)
5. attributes (could be blank)
6. locus ID/assembly ID
7. genome ID

An example is below

```
1 1 baseplate assembly protein 0 SK57_RS00005 BIDMC87
2 1 baseplate assembly protein 0 SK57_RS00010 BIDMC87
7 1 major tail tube protein 0 SK57_RS00035 BIDMC87
8 1 mu-like prophage FluMu gp41 family protein 0 SK57_RS00040 BIDMC87
```

3.3 Output Directory structure

PanACEA splits the output into several sub-directory to allow for easier searching if the user would like examine individual files. If the user has entered OUTDIR and NAME to the program the directory structure is:

```
OUTDIR
└── NAME
    ├── main.html
    ├── core
    │   ├── Core_Region.1.SVG
    │   └── Core_Region.2.SVG
    ├── fgi
    │   ├── CL_INS_1.html
    │   └── CL_INS_2.html
    ├── json
    │   ├── CL_1.allFasta.json
    │   └── CL_2.allFasta.json
    └── scripts
        ├── fgi.functions.js
        ├── geneFile.functions.js
        └── main.javascript.js
```

3.4 Phylogeny and Metadata files

PanACEA can read in a tree file which will then be used in images at the FGI, Core Region, and Gene page view. PanACEA requires the tree to be in Newick format and the Metadata file to be in a tab-delinated format with three columns. Column 1 is the Genome ID, column 2 is the metadata variable name, and column 3 is the value. For example

```
GenomeA outcome lived
GenomeB outcome died
GenomeC outcome lived
```

3.5 Configure Files

The PanACEA can use two config files: (1) The Functional Annotation data file that is used to add functional annotations and colors to the figure and (2) the Output Configure file that allows user to change the size and shape of the output image. Example config files for both are included in the download space.

3.5.1 Functional Annotation File

The functional annotation config file is modular with different annotation type each having it's own section. Each section is started with a line containing "START" and ends with a line contatining only "END". In between these, the lines should be have the variable name followed by a space and then followed by the value for one of four different variables:

1. **ID**: a string used on the table tab
2. **mapfile**: a tab-delineated file with gene or centroid ID in the first column and a comma-separated term list in the second
3. **ontology**: an obo type onotology file containing definitions for the terms in the mapfile
4. **name**: a tab-delineated file with gene or centroid ID in the first column and a function in the second, OR a string that is assigned to each of the genes in the mapfile
5. **color**: a tab-delineated file with the function in the first column and a color in the second, OR a rgb color string that is assigned to each of the gene with the funciton

The functional assignment of the first modules are prioritized over following modules. An example of a module is below:

```
START ID AROTerms
mapfile /home/tclarke/pan_html/ARO.mapterm.txt
ontology /usr/local/db/card/card_current/aro.obo
name Antibiotic Resistance
color ff0000
END
```

Of a mapfile:

```
centroid_7381 ARO:3002676,ARO:3002679,ARO:3002670,ARO:3002675
centroid_2818 ARO:3003392
centroid_6236 ARO:3002707
```

Of a name file:

```
centroid_12202 Other
centroid_2996 Regulatory functions
centroid_21112 Transport & binding proteins
centroid_17893 Other
```

And a color file:

```
Regulatory functions 0000ff
Biosynthesis of cofactors, prosthetic groups & carriers 87ceff
Cellular processes a0fc8d
Mobile & extrachromosomal element functions 8e0000
```

Included in the PanACEA package are two perl scripts designed to facilitate annotating the PanOCT genes in the PanACEA viewer. The first is make_rgi_clusters.pl, which translates RGI's output to a PanACEA readible format. It's command line is as follows:

- **-d** input RGI dataSummary.txt file. Required
- **-o** output file. Default is "aro_centroid.list.txt" current directory.
- **-t** shows "best" (default) or "all" RGI hits. If neither of these are inputed, uses "best"

The second script is make_conf_file.pl which reads a PanOCT output directory and generates the annotation configure file:

- **-d** input panoct output directory
- **-o** output file

3.5.2 Graphic Annotation File

The Graphical Annotation config file allows the user to change the dimensions and colors of the image. The file consists of unique lines with the variable name followed by a space and then followed by the value for one of N different variables. Variable names not listed below will be ignored. Repeated Variables will be given the last value. A list of the variables along with their default value and a description are as follows:

1. **BACKGROUND**: ("E0E0E0") color of the background of the outer ring/core regions on the main view
2. **fGR**: ("FF0000") color of the fGRs in the outer ring/core regions on the main view

3. **BORDER_SIZE**: (120) size of the border between the outer ring and the edge of the circle in pixels

4. **CIRCLE_SIZE**: (400) size of the radius of the outer edge of the outer ring in pixels

5. **TEXT_SIZE**: (6) size of the text in pixels

6. **GENE_SIZE**: (24) size of the gene in pixels

The total size of the image is 2 x (BORDER_SIZE + CIRCLE_SIZE) with the pan-chromosome being 2 x (CIRCLE_SIZE). An example file showing the default values is below:

```
BACKGROUND E0E0E0
fGR FF0000
BORDER_SIZE 120
CIRCLE_SIZE 400
TEXT_SIZE 6
GENE_SIZE 24
```

4 Browsing through PanACEA

Once the Perl script has been run, the images have all been created, the scripts written, and the pan-chromosomes ready to be explored. The main html page address is given by the script at the end of the run. Using this address in a web browser will load up the main screen, though the main page may take a while to load. We have designed PanACEA to work on multiple web browsers and it was tested on Explorer, Chrome and Firefox. The following images are all taken from a use on Firefox on an Enterobactor run with GO Terms and Antibiotic Resistance genes.

4.1 Main HTML page

The main page gives a summary image of either an pan-chromosome or an cycle-fGRs. The pan-chromosome summary image contains a an outer ring showing the different regions (A) and a core gene view with the core genes on the positive strand on the middle ring (B) and the negative strand on the inner most ring (C). If the pan-chromosome has been assembled as a closed chromosome, the circle is complete. If it is not, a small break at the very top of the chromosome is shown (D shows the location where the gap would be).

In the center of the image is a circle. The top of the circle contains a legend (E) showing the colors associated with each of the functional categories provided to the Perl script and two disk icons (F) allowing the user to save the rings and the legend at the current visible stat either as a PNG or a SVG. The bottom of the circle contains the table (G). At loading the table is hidden and can be accessed by clicking the "Show Table" button (H). On the far left is the Assembly Pan-Chromosome side menu bar (I).

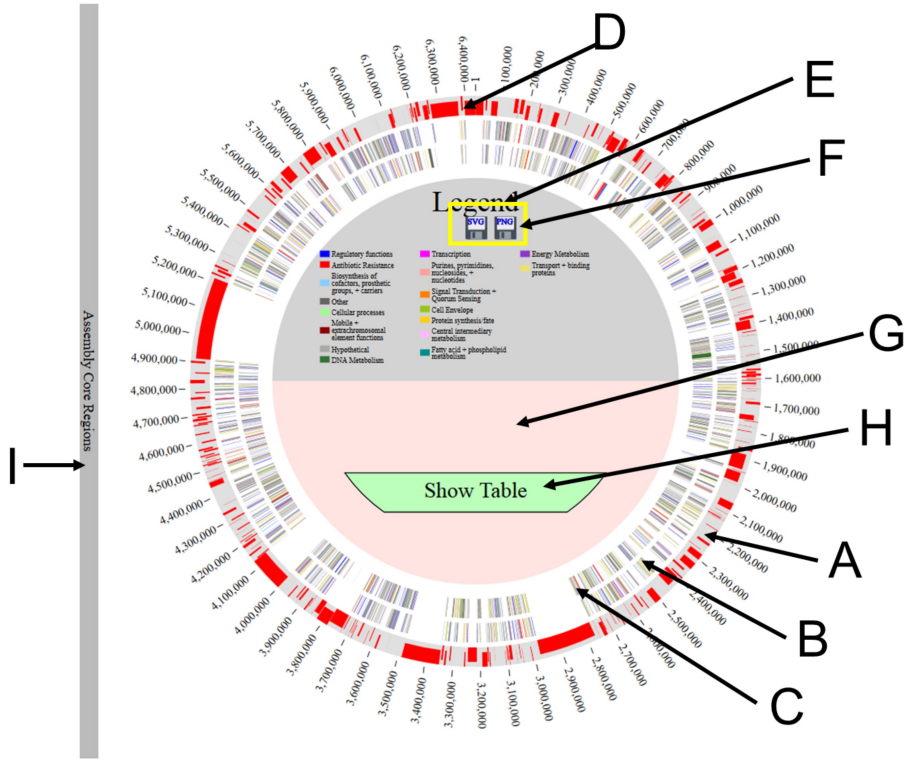


Figure 1: Main Page Pan Chromosome View

For cycle-fGRs, only genes are shown. Since these regions have been classified as circular, only the circle view is shown. The genes are shown around the edge with the direction shown with arrowhead ends, and the color is the functional assignment (A). Clicking the gene will show the gene page.

Since there are numerous images that are loaded into the main page, the The PanACEA browser can take up to several minutes to load. To give the user that the script is working, a loading screen of a rotating expanding and shrinking JCVI four square logo is shown until the screen is fully loaded. After the page is loaded and the page is doing calculations , the main screen will fade slightly and a circle will move around the central legend and table area. During this time, the user will be unable to click anything.

4.1.1 Pan-Chromosome Region View

The outer ring differentiates between two types of regions, the core and the flexible genomic (fGRs). The core regions are shown in light gray (A) and the fGRs are shown in red (B). The fGRs are staggered vertically for easier visual detection of neighboring regions. Whether a fGR is either "up" or "down"on the

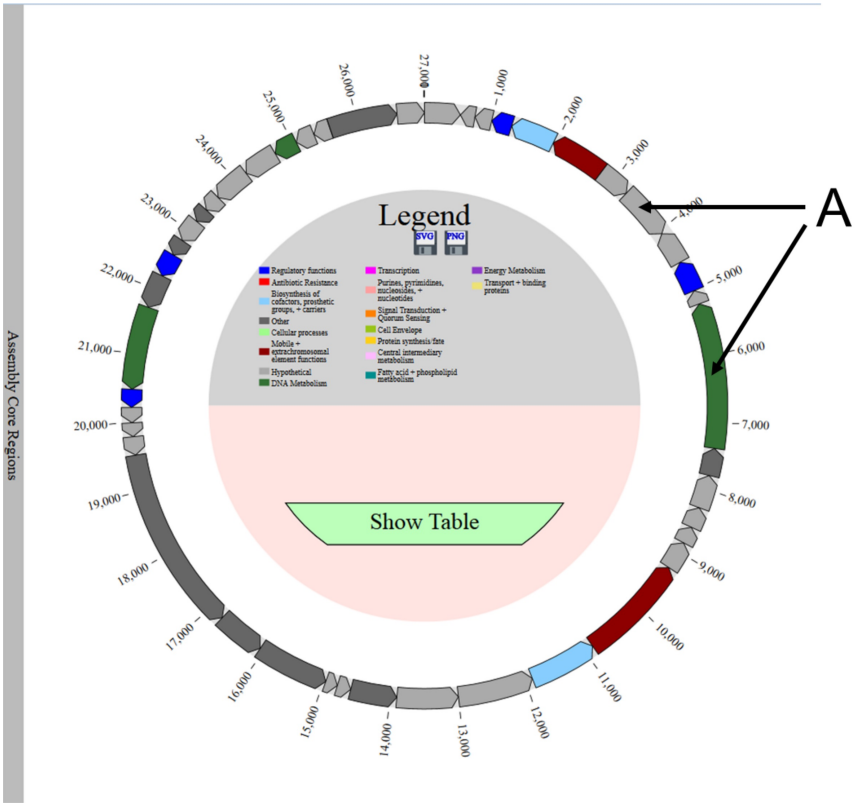


Figure 2: Main Page Circular FGR view

ring has no significance. Moving a mouse over a region will change the color of the region to dark gray (C) and display a graphical preview of the region in the closest corner (D) while clicking on the region will open the detail region view in a separate window. The regions preview formats are also different. The Core Region preview is shown using the exact same iconography as the Core Region region page. The fGRs are shown as a reduced version of its detailed view, with the genes shown as colored circles in different rows showing the different arrangements (E). The number of genomes with each arrangement is shown on the right edge of the row.

4.1.2 Using the Legend

The legend allows users to highlight regions and genes by different high level functions. Functions can be selected by clicking on the block or name (A). This will turn on the function which will be shown by a black outline around the color in the legend (B). The regions are highlighted based on whether the region contains at least one gene cluster that has been annotated with each

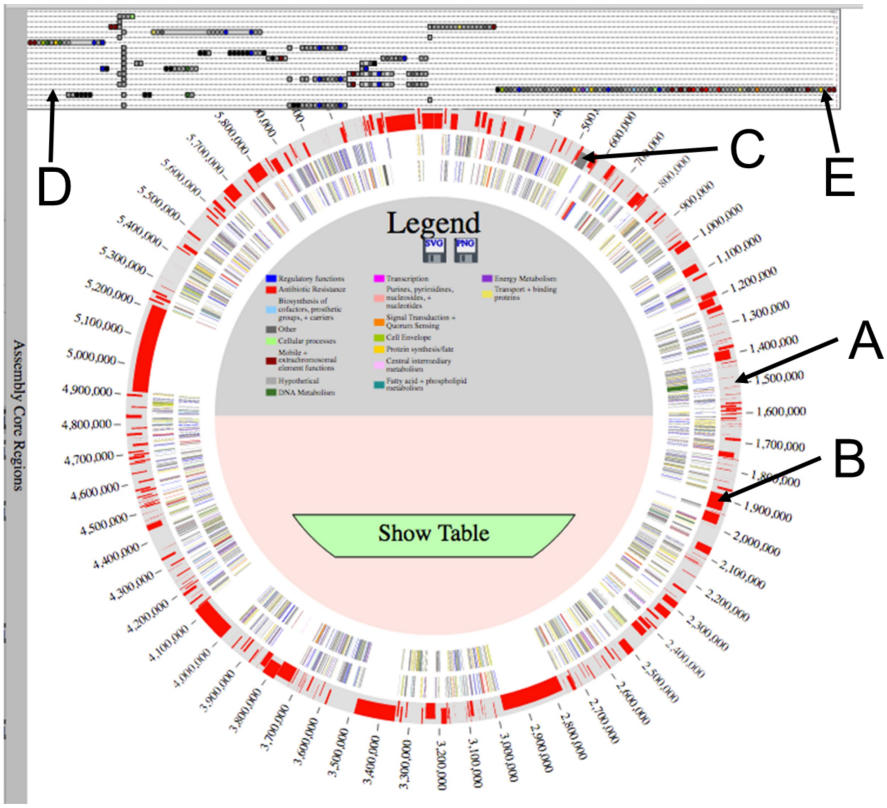


Figure 3: Main Page Region Preview

selected function. Highlighted core regions are outlined in black (C), while all non-highlighted regions are not. All the core genes within highlighted regions regions containing a selected function are also kept on their color while the rest are dimmed. Furthermore, the table is reduced to the rows that are connected to the selected functions.

4.1.3 Using the Table

The table contains two built in tabs: the regions and the genes. Also, any functional information obtained from a role and an ontology file from the functional config file is also added. Once a table is shown using the button described above, button changes to a "Hide Table Button" (A) with the table shown above the button (B). The information within the table can be changed by using the tabs below (C) to show either information about the genes or regions, or if ontologies have been added, genes and regions with the ontology (D). Selecting a row with a region will act similarly to a mouse-over of that region. Selecting a gene will highlight that gene and the region containing the gene. Selecting an functional

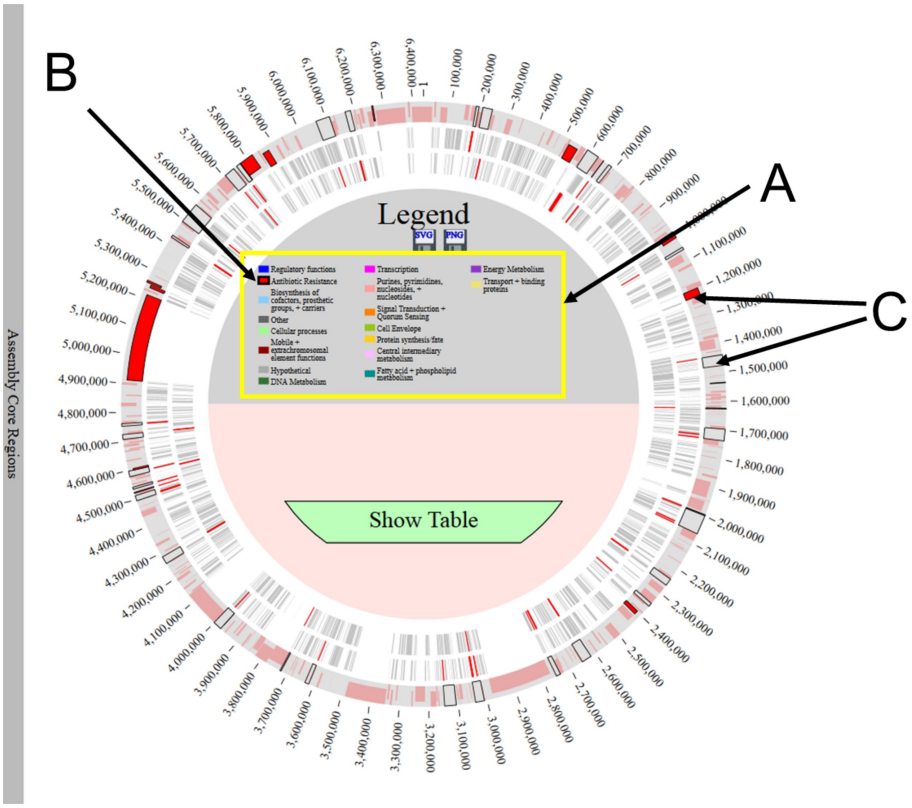


Figure 4: Main Page Legend

category will perform the same tasks as selecting a functional in the legend. The function in the legend most associated with the term when availible will also be highlighted. While a term is highlighted in the table, no function button in the legend is active.

4.1.4 Selecting a different Pan-Chromosome

In runs where multiple pan-chromosomes were generated, the user can select any of those, in addition to any circular FGRs. Both are listed in the left-most menu when can be accessed by clicking the bar reading "Assembly Core Regions" (A). The assembly core menu consists of thumbnails of the cores and cycle fGIs along with their numerical ID in the center. Each can be selected by clicking on the thumbnail. This will change the main html view to that.

The menu contains both assembly cores and cycle fGIs. The assembly core are the pan-chromosomes containing both FGIs and core genes, where the cycle FGRs are the regions that are circular, i.e. those where the order loops back to the beginning. To view additional thumbnails, the menu can be scrolled down

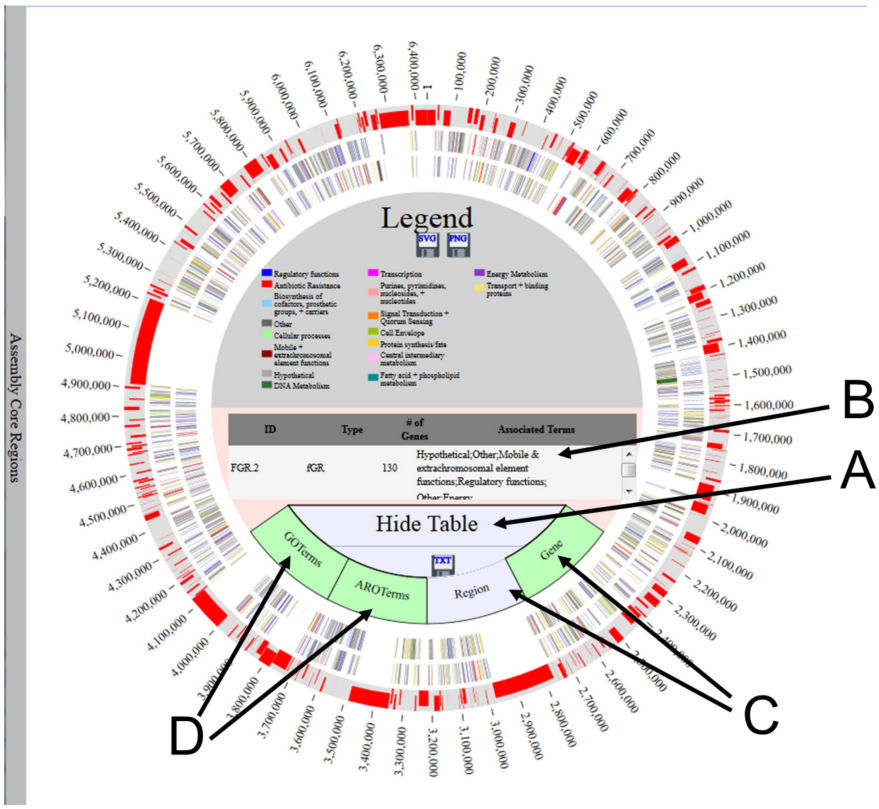


Figure 5: Main Page Table

using the scroll bar. Also, all the thumbnails can be hidden by clicking on the menu header (B).

4.2 Core Region View

The core region view consists of two horizontal window above a panel showing the phylogeny when available. The top window (A) shows the gene model with the arrow showing the direction, the width the length of the gene and the color the function. The bottom window (B) shows a histogram where the height of each bar is the number of genomes containing the gene, from all from all the genomes (highest possible bar) to the minimum core rate set by the user and calculated during the Perl run (x-axis). Moving the mouse over the gene models in the top window will show the gene name and the mean length and standard deviation of the gene length across the genomes. Moving the mouse over the bars in the bottom window will show the gene name and the number of genome. Clicking on top window will load the gene page for the given gene.In instances where the Core Region is wider than available window, scroll bars are shown.

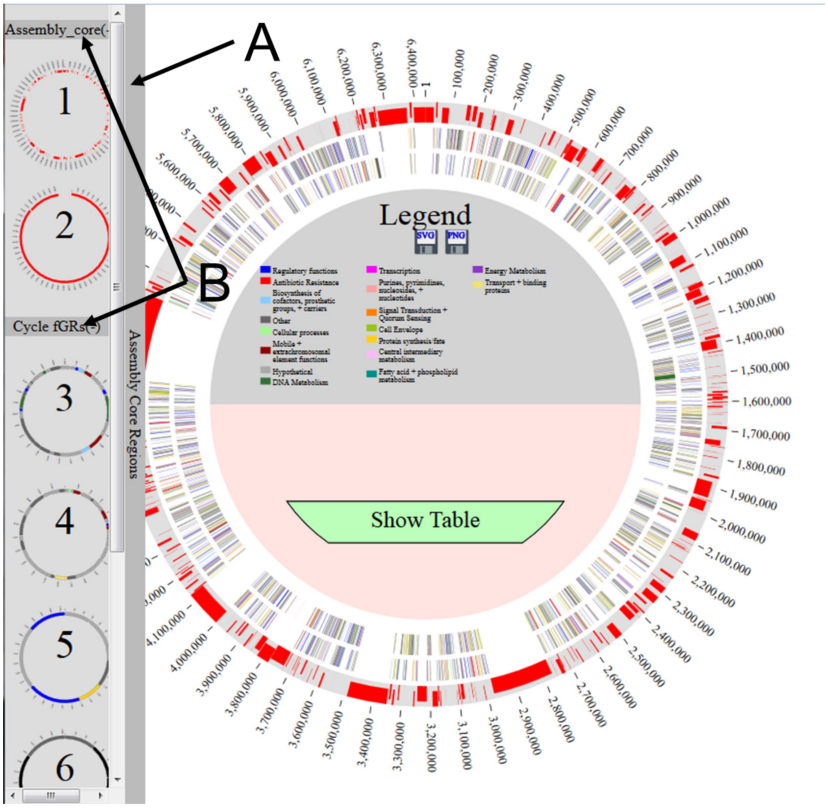


Figure 6: Main Page Changing Chromosome View

Beaneah the window is the legend (C), which like the legend in the Main and fGR view, shows the colors and associated functions. Below the legened is a functional set of buttons that allows the user to zoom in and out of the core region view (D). This however, does not change the view window.

At the bottom of the page is a drawing of the phylogeny when a user supplies one. The panel is divided into 6 different sections, shown in clockwise from the top-left below. (E) is the tree panel where the tree is shown. All metadata annotation that the user has supplied is shown on the outer ring. (F) is the tree viewer control panel where the user can change the style of tree shown (linear or circular) or the depth of the tree (tree level where 1 is the full tree including all the leaves).) The users can also save the tree as a picture (PNG or SVG).(G) is the node information viewer that includes the genomes that descend from this node. (H) shows the percentage of descending nodes. Both of these are explained in more detail below. (I) is the metadata control panel, where users can select to add or remove metadata information to the tips of the trees. (J) is the legend box where the colors corresponding to different metadata values

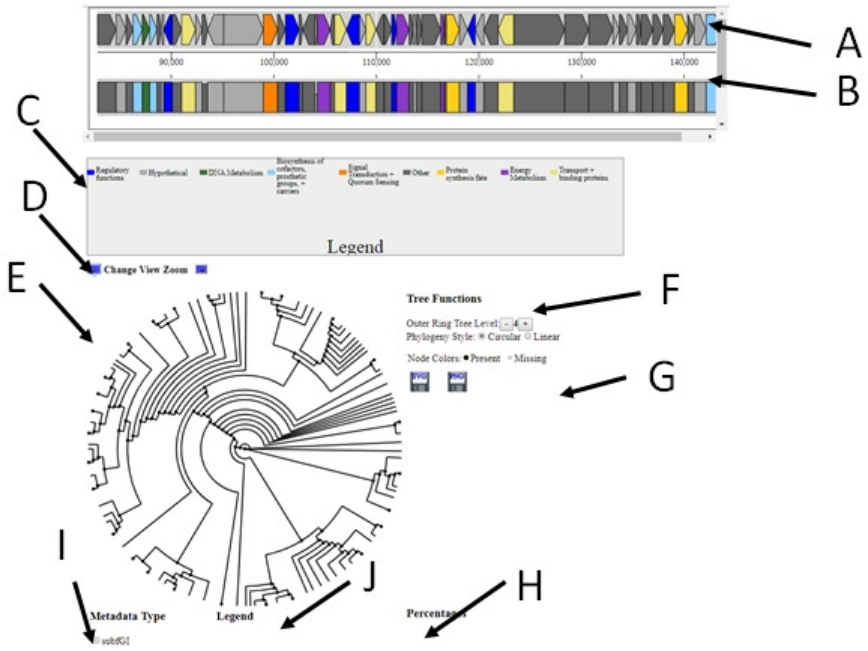


Figure 7: Core Region View

are shown.

Clicking on the histogram on the bottom window will highlight the selected fGR (A) and label the phylogeny based on the pattern of genomes that contain the fGR. Nodes where all the descendents contain the gene are colored black (B) while nodes with no descendents containing the gene are shown as a light gray circle (C). Nodes that have both are shown as a pie chart with the black slice indicating the percentages of decedents containing the fGR (D).Moving the mouse over a node will turn it "On" by doubling the radus of circle and coloring it as green (E). The genomes that descend from this node are shown in the mid-left box (F). If the metadata and fGRs are selected, then the percentage of the nodes that contain the fGR and belong to the metadata are shown in the cell labeled percentages (G). Moving the mouse out of the circle will turn it "Off" and empty the genomes and percentages boxs and return the node to its previous shape and size. However, clicking on the node will keep it "On" even if the mouse is moved out.

4.3 Flexible Gene Region

The flexible region page contains a main central window (A) showing the different gene components of the different regions that is flanked the core genes, or if the assembly has a break, an empty gene space shown in an outline with

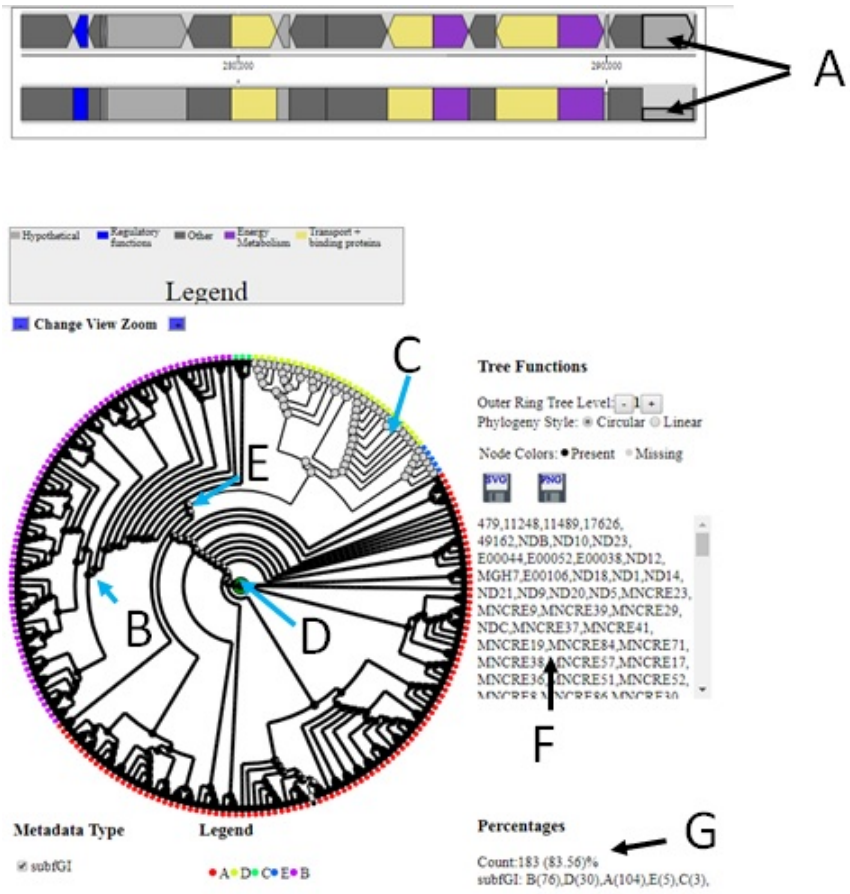


Figure 8: Core Gene Phylogeny View

dotted lines (B). Each row in the central window shows a unique arrangements of genes with the one gene per column. The gene ID is shown on the bottom row (C) and the flexible gene island ID is shown on the top row (D). The genes are again colored by their functionality with the arrow pointing in the direction of coding. Clicking on the gene icon or on the ID will load the gene page. To allow for easier analysis, breaks in fGIs are shown by a change in background color using dark and light gray. Also vertical lines have been drawn for each gene. For flexible gene regions where the number of genes or the number of gene arrangements exceeds the window size, the window can be scrolled both horizontally and vertically.

The right flanking window contains several tools to assist in the analysis of the window. Each unique gene arrangement has five functions (from left to right): a highlight row button (A), a select row box (B), the flanking gene icon

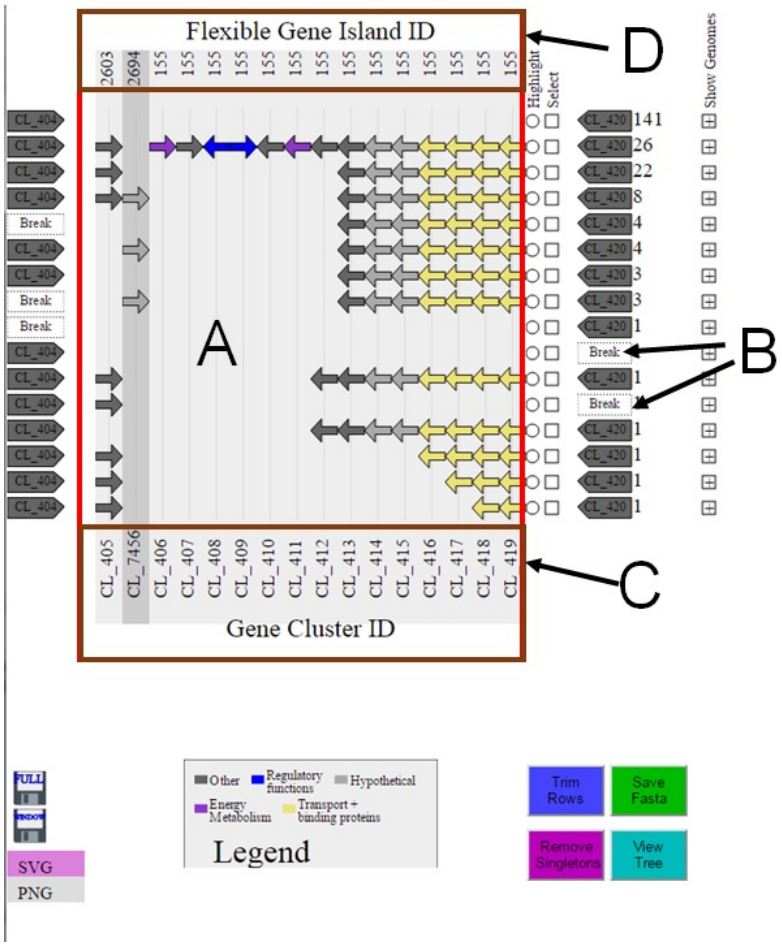


Figure 9: fGR Main View

(C), the genome count (D), and the display genome button (E). The highlight button colors the row in a bright yellow. Only one row can be highlighted at a time. The select row button allows the user to choose rows to keep upon clicking the Trim Rows (F) button below (explained in more detail later). The flanking gene icon shows the gene ID, direction, and function of the core gene bounding the flexible gene region. The genome count shows the number of genomes containing this unique gene arraignment. To see a list of the genomes, the user can click on the display genomes icon, and a list of genome IDs will appear beneath the gene icon (G). This will also vertically expand the row in the central window, but only one copy of the genes will be shown in the standard height in the row. If the user scrolls vertically, the genes will remain in the visible row box if necessary.

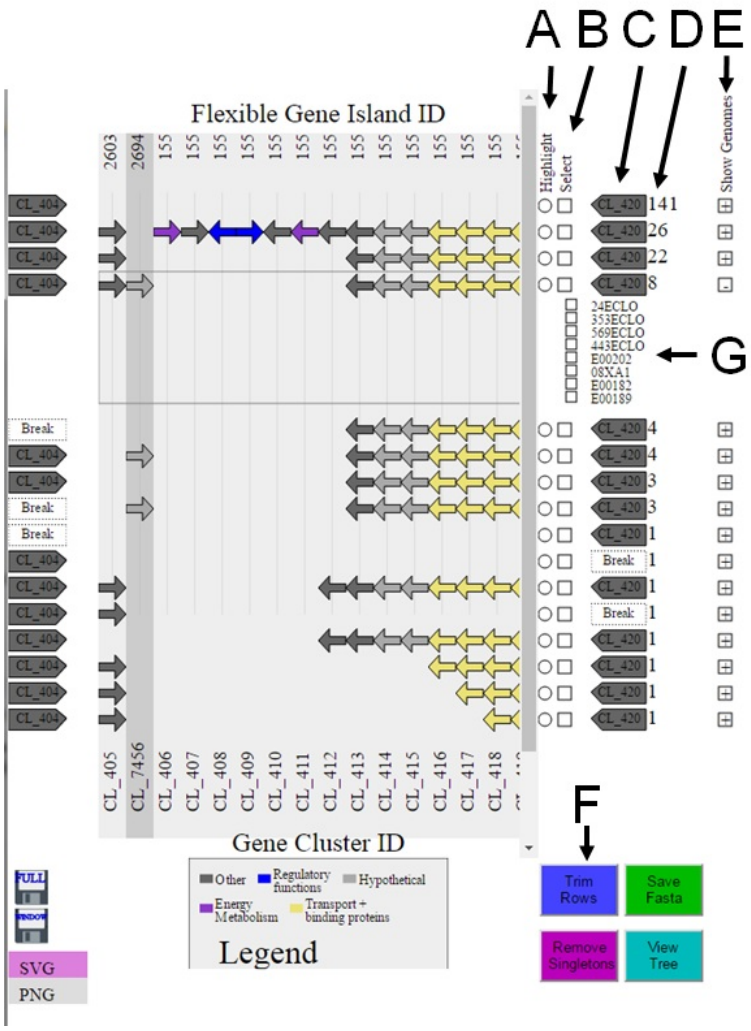


Figure 10: fGR Boundary Regions

The viewer also contains a footer with three boxes. The central footer box contains the legend with the functional colors in the window enumerated (A). The left box contains two icons to save the image, either the "Full" image (B) which saves the complete Flexible Gene Region, including those genes and arrangements that are currently hidden, or the "Window" (C) view which only saves the genes and arrangements currently in the viewer window, including the bounding core genes. The selection boxes beneath the icons (D) give the user the option to save the image either as a SVG (default) or a PNG, with the selected option shown in pink. The right box gives the window tool functions.

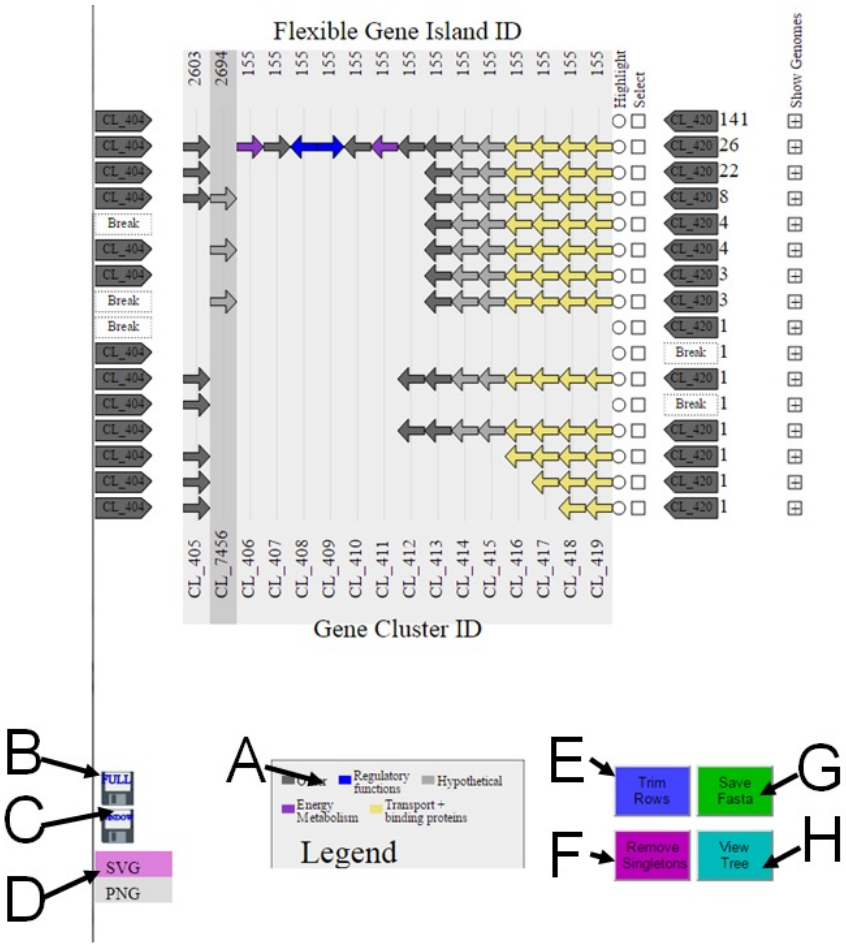


Figure 11: fGR Sort and Save Tools

The top right button, Trim Row (E), allows the user to remove all rows that the user has not selected. This will also remove all columns that does not contain a gene in the arrangements contained in the selected rows. The bottom right button, Remove Singletons (F), removes all the arrangements seen in only a single genome. After selecting, the button will then let user restore the previous image. Currently, it does not remove the empty columns. After clicking it, the button changes to "Add Singletons" and clicking it will restore the singletons. The top left button is "Save Fasta" (G) and this let users save a multiple fasta file consisting of each FGI for each genome. When a phylogeny has been included, a button at bottom left, "Show Tree", will appear, and clicking on it will open a new page, showing the location of any selected FGI on the phylogeny. If no

FGIs are selected, an error message occurs.

4.3.1 fGR Phylogeny View

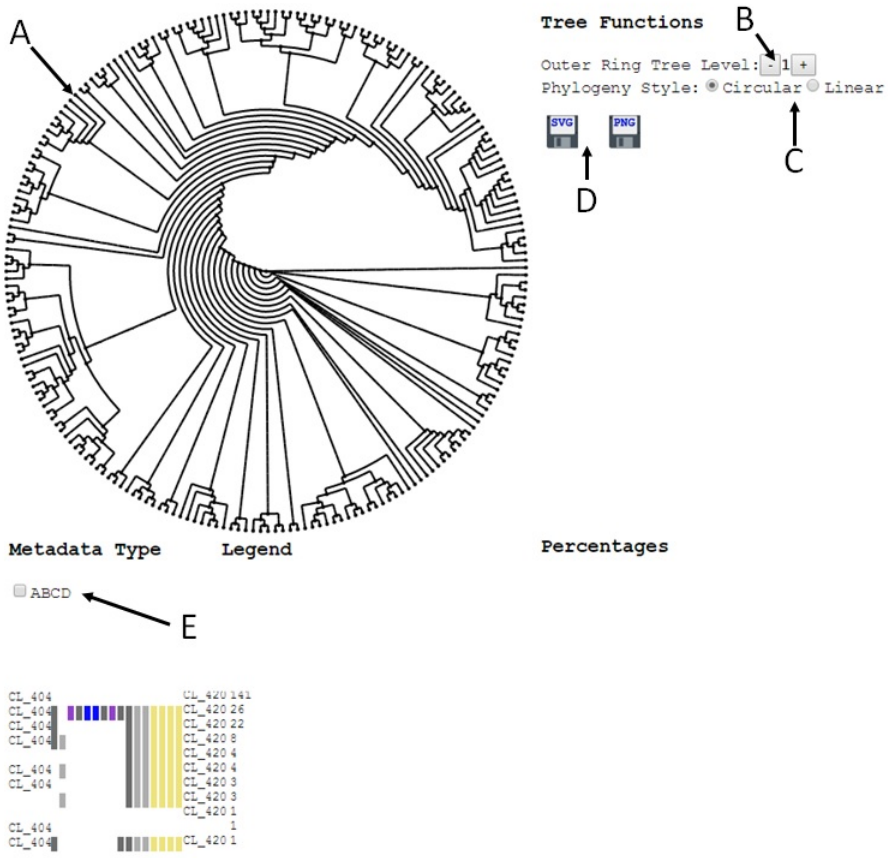


Figure 12: Phylogeny View

The fGR tree figure allows the user to observe the evolutionary pattern of retention of a selected set of fGIs. The phylogeny is shown on top left (A) and can be varied with the functions shown on the top right: changing the level of the outer ring (B), where level with the leaves is 1; changing the shape of the phylogeny (C); and saving the phylogeny images as a PDF or an SVG (D). Below the phylogeny is the Metadata Annotation (E).

Below the phylogeny are all the selected fGRs. Clicking on a fGR will highlight the selected fGR (A) and label the phylogeny based on the pattern of genomes that contain the fGR. Nodes where all the descendents contain the fGR are colored black (B) while nodes with no descendents containing the fGR are

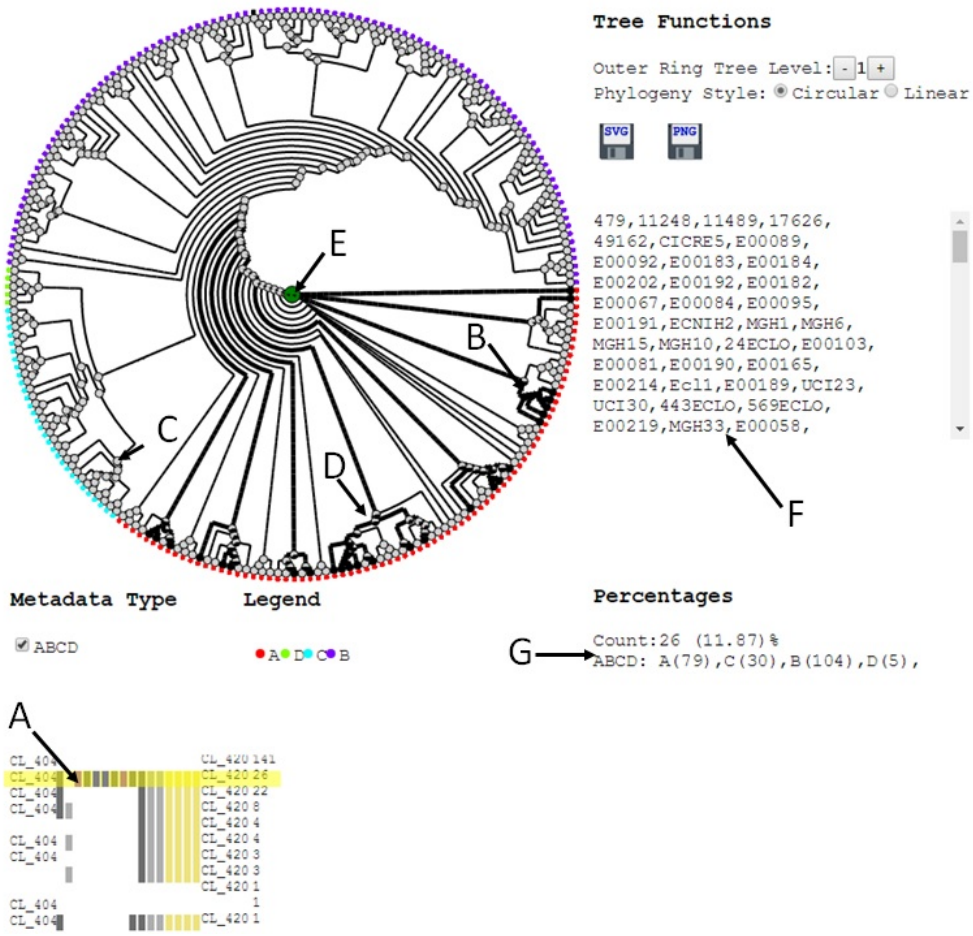


Figure 13: View of fGR on Phylogeny

shown as a light gray circle (C). Nodes that have both are shown as a pie chart with the black slice indicating the percentages of decedents containing the fGR (D).Moving the mouse over a node will turn it "On" by doubliing the radus of circle and coloring it as green (E). The genomes that descend from this node are shown in the mid-left box (F). If the metadata and fGRs are selected, then the percentage of the nodes that contain the fGR and belong to the metadata are shown in the cell labeled percentages (G). Moving the mouse out of the circle will turn it "Off" and empty the genomes and percentages boxs and return the node to its previous shape and size. However, clicking on the node will keep it "On" even if the mouse is moved out.

4.4 Gene Page

Cluster	CL_5813
Name	Aac(6)-IIc
Region	FGI_447
# of Genomes	16
Functional IDs	Antibiotic Resistance
Associated Terms	ARO:3002596(AAC(6)-IIc) GO:0008080(N-acetyltransferase activity) GO:0016747(transferase activity, transferring acyl groups other than amino-acyl groups)
Mean	652
Standard Deviation	
Minimum Length	
Maximum Length	
Sequence	ATGTCCGCCAACAATGCCGCAATAGTTCTACGAGTCATGGCCGAGAACGATCTGCCAATG CTCCATGCTTGGCTGAACCGCCCCCACATAGTCGAGTGGTGGGGCGGCGAGGATGAACGC CCAACTCTTGACGAAGTCTTAGAACACTATTCGCCCGAAGTTCTGGCAAAGCAAGCTGTA GTGCCTTACATCGCAATGCTAGATGACGAACCCATCGGCTACGCCCAATCCTACATCGCA CTTGGAAGTGGCGATGGATGGTGGGAAGACGAAACTGATCCAGGGGTCCGCGGGATTGAC CAGTCTTTGGCTAATCCATCACAGTTAAACAAGGGGTTGGGTACAAAGCTCGTACGCTCG CTCGTTGAACTCCTGTTTAGCGACCCGGCCGTAACGAAAATCCAAACCGATCCATCTCCT AGCAACCATCGCGCCATTCGCTGCTACGAGAAGGCCGGGTTCGTTCAAGAAAAAAACATC CTCACACCTGACGGCCCTGCGGTGTACATGGTCCAAACACGCCAGGCGTTCGAAAGCCTG CGCACTGTTCAAAGCTTCAAAATCAAGGGGAAGTGGTCATGA
Genomes	E00184,E00209,MNCRE19,E00001,MGH14,E00050,E00054,E00197,202ECLO,MGH3,E00202,E00065,E00192, BH10892,E00043,E00183

Figure 14: Gene Page Summary

Each gene whether selected on the main page, the core region or the flexible gene region has the same style page. Currently, three possible views are shown that can be selected using the radio buttons on the top (A). For the first two, only the selected view is shown on the screen. The first is the summary or table view (B) which contains a list containing: the cluster ID, the gene name, the region where the gene is found, the number of genomes that contain the gene, the single high level function assigned to the gene, the multiple low level functions, length information including the mean, standard deviation, minimal and maximum length, the centroid sequence and a list of genomes.

Figure 15: Gene Page Fastas

The second view is the Centroid Fasta (A) which shows the centroid sequence in fasta format. The final button is an option to download all the sequences in fasta format. If the user has created multiple alignments, these will be downloaded. Otherwise, the non-aligned fasta sequences will be. Given

that the sequences are downloaded, the current view, either the summary or cetroid fasta, will remain.

Other view options are the Multi-Fasta viewer powered by MSA Viewer (shown in http://msa.biojs.net/) and the gene phylogeny view. The phylogeny view is the same layout as the fGR viewer, with the nodes automatically colored by whether the desendents contain the gene.