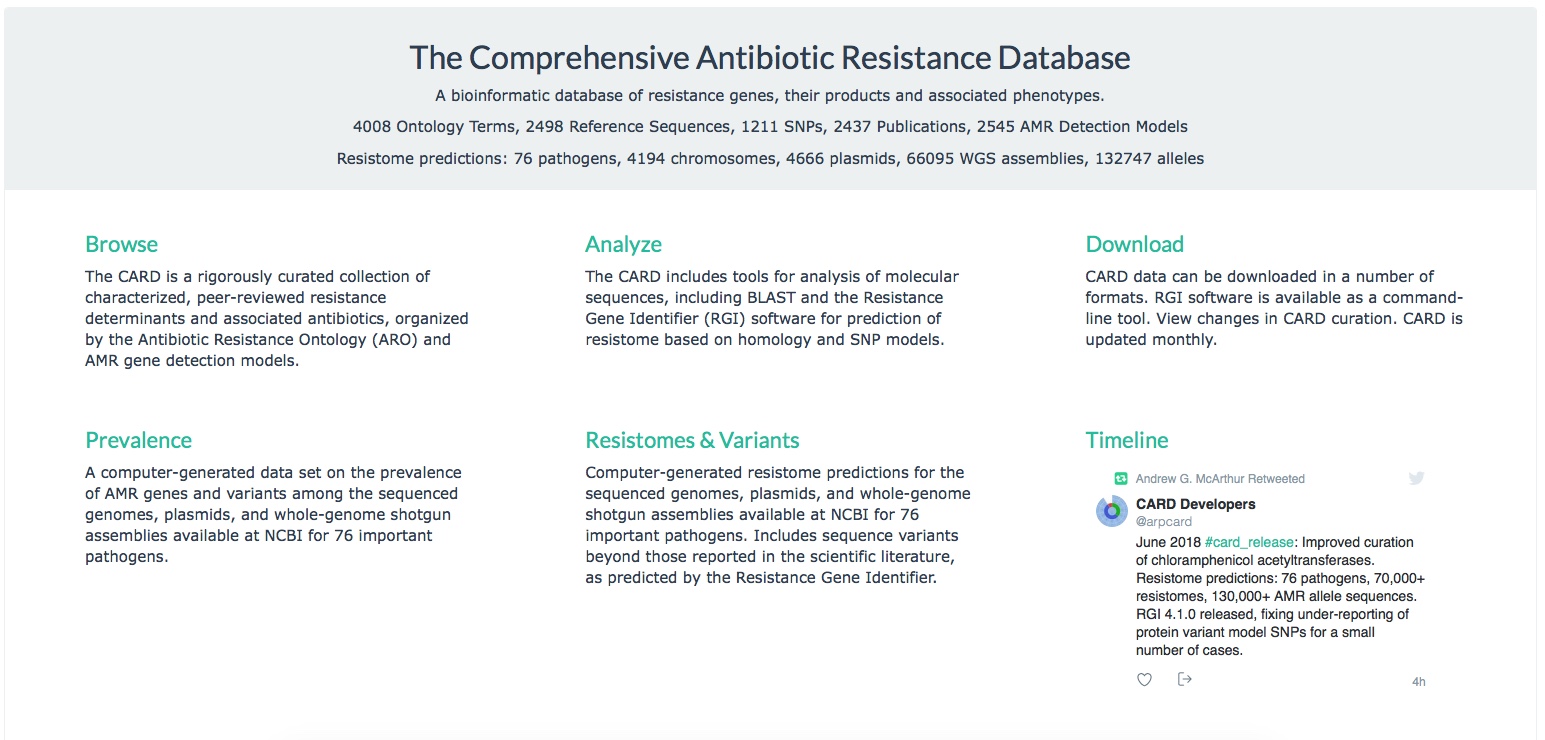
****

**Introduction**

This module gives an introduction to prediction of antimicrobial resistome and phenotype based on comparison of genomic or metagenomic DNA sequencing data to reference sequence information. While there is a large diversity of reference databases and software, this tutorial is focused on the Comprehensive Antibiotic Resistance Database (CARD) for genomic AMR prediction.

**From Analysis to Interpretation**

The relationship between AMR genotype and AMR phenotype is complicated and no tools for complete prediction of phenotype from genotype exist. Instead, analyses focus on prediction or catalog of the AMR resistome – the collection of AMR genes and mutants in the sequenced sample. While BLAST and other sequence similarity tools can be used to catalog the resistance determinants in a sample via comparison to a reference sequence database, interpretation and phenotypic prediction are often the largest challenge. To start the tutorial, we will use the Comprehensive Antibiotic Resistance Database (http://card.mcmaster.ca) to examine the diversity of resistance mechanisms, how they influence bioinformatics analysis approaches, and how CARD’s Antibiotic Resistance Ontology (ARO) can provide an organizing principle for interpretation of bioinformatics results.



CARD’s website provides the ability to:

1. Browse the Antibiotic Resistance Ontology (ARO) and associated knowledgebase.
2. Browse the underlying AMR detection models, reference sequences, and SNP matrices.
3. Download the ARO, reference sequence data, and indices in a number of formats for custom analyses.
4. Performed integrated genome analysis using the Resistance Gene Identifier (RGI).

In this part of the tutorial, your instructor will walk you through the following use of the CARD website to familiarize yourself with its resources:

1. Examine the mechanisms of resistance as described by the Antibiotic Resistance Ontology.
2. Examine the NDM-1 beta-lactamase protein, it’s mechanism of action, conferred antibiotic resistance, it’s prevalence, and it’s detection model. [BLASTP of NDM-1 against CARD]
3. Examine the aac(6')-Iaa aminoglycoside acetyltransferase, it’s mechanism of action, conferred antibiotic resistance, it’s prevalence, and it’s detection model. [BLASTP of aac(6')-Iaa against CARD]
4. Examine the recently reported colistin resistance MCR-1 protein, it’s mechanism of action, conferred antibiotic resistance, it’s prevalence, and it’s detection model. [BLASTP of MCR-1 against CARD]
5. Examine the fluoroquinolone resistant gyrB for *M. tuberculosis*, it’s mechanism of action, conferred antibiotic resistance, and it’s detection model. [Why would BLASTP be inappropriate for this resistance determinant?]
6. Examine the glycopeptide resistance gene cluster VanA, it’s mechanism of action, conferred antibiotic resistance, and it’s detection model(s). [Why would BLASTP be inappropriate for this resistance determinant?]
7. Examine the MexAB-OprM efflux complex, it’s mechanism of action, conferred antibiotic resistance, it’s prevalence, and it’s detection model(s). [Why would BLASTP be inappropriate for this resistance determinant?]

**RGI for Genome Analysis**

As illustrated by the exercise above, the diversity of antimicrobial resistance mechanisms requires a diversity of detection algorithms and a diversity of detection limits. CARD’s Resistance Gene Identifier (RGI) currently integrates four CARD detection models: Protein Homolog Model, Protein Variant Model, rRNA Variant Model, and Protein Overexpression Model. Unlike naïve analyses, CARD detection models use curated cut-offs, currently based on BLAST/DIAMOND bitscore cut-offs. Many other available tools are based on BLASTN or BLASTP without defined cut-offs and avoid resistance by mutation entirely.

In this part of the tutorial, your instructor will walk you through the following use of CARD’s Resistome Gene Identifier with “Perfect and Strict hits only”:

1. Resistome prediction for the multidrug resistant *Acinetobacter baumannii* MDR-TJ, complete genome [NC\_017847]
2. Resistome prediction for the plasmid isolated from Escherichia coli strain MRSN388634 plasmid [KX276657]
3. Explain the difference in triclosan resistance between two clinical strains of *Pseudomonas aeruginosa* that appear clonal based on identical MLST [Pseudomonas1.fasta, Pseudomonas2.fasta] (these are SPAdes assemblies)

**RGI at the Command Line**

RGI is a command line tool as well, so we’ll do a demo analysis of 112 clinical multi-drug resistant *E. coli* from Hamilton area hospitals, sequenced on MiSeq and assembled using SPAdes. We’ll additionally try RGI’s beta heat map tool (planned release August 2018).

Login into your course account’s working directory, make a module 4 directory, and set some aliases for this demo:

mkdir module4

cd module4

alias rgi="python3 /home/ubuntu/CourseData/IDGE\_data/module4/rgi-4.1.0/rgi"

alias rgiviz="python3 /home/ubuntu/CourseData/IDGE\_data/module4/repository/heatmap/snsheatmap.py"

Take a peak at the list of *E. coli* samples and the options for the RGI software:

ls /home/ubuntu/CourseData/IDGE\_data/module4/ecoli

rgi -h

First we need to acquire the latest AMR reference data from CARD:

rgi load -h

wget https://card.mcmaster.ca/latest/data

tar -xvf data ./card.json

less card.json

rgi load -i card.json --local

ls

We don’t have time to analyze all 112 samples, so let’s analyze 1 as an example (the course GitHub repo contains an EXCEL version of the Ecoli\_37.txt file):

rgi main –h

rgi main -i /home/ubuntu/CourseData/IDGE\_data/module4/ecoli/Ecoli\_37.fasta -o Ecoli\_37 -t contig

-a BLAST -n 4 --local --clean

ls

less Ecoli\_37.json

less Ecoli\_37.txt

I have pre-compiled results for all 112 samples, so let’s try RGI’s beta heat map tool (pre-compiled images can be downloaded from the course GitHub repo):

ls /home/ubuntu/CourseData/IDGE\_data/module4/ecoli\_json

rgiviz –h

rgiviz -i /home/ubuntu/CourseData/IDGE\_data/module4/ecoli\_json -category gene\_family

-o genefamily\_samples -cluster samples

rgiviz -i /home/ubuntu/CourseData/IDGE\_data/module4/ecoli\_json -category drug\_class

-o drugclass\_samples -cluster samples

rgiviz -i /home/ubuntu/CourseData/IDGE\_data/module4/ecoli\_json -o cluster\_both -cluster both

rgiviz -i /home/ubuntu/CourseData/IDGE\_data/module4/ecoli\_json -o cluster\_both\_frequency

-frequency on -cluster both

ls

**RGI for Merged Metagenomics Reads**

The standard RGI tool can be used to analyze metagenomics data, but only for merged reads with Prodigal calling of partial open reading frames (ORFs). This is a computationally expensive approach, since each merged read set may contain a partial ORF, requiring RGI to perform massive amounts of BLAST/DIAMOND analyses. While computationally intensive (and thus generally not recommended), this does allow analysis of metagenomic sequences in protein space, overcoming issues of high-stringency read mapping relative to nucleotide reference databases. Assembled metagenomic data could alternatively be used instead of merged reads.

Lanza et al. (*Microbiome* 2018, 15:11) used bait capture to sample human gut microbiomes for AMR genes. Using the online RGI under “Low quality / coverage” and “Perfect, Strict and Loose hits” settings, analyze the first 500 merged metagenomic reads from their analysis (file ResCap\_first\_500.fasta). Take a close look at the predicted “sul2” and “sul4” hits. How good is the evidence for these AMR genes? (**Don’t use the “AMR Genes” visualization, it has a bug for multiple hits to the same gene**)

**Metagenomics Data and the Burrows-Wheeler Transform**

The most common tools for metagenomic data annotation are based on high-stringency read mapping, such as the Burrows-Wheeler Transform (BTW). Available methods almost exclusively focus on acquired resistance genes, not those involving resistance via mutation. CARD and other AMR reference databases include sequences and mutations from the published literature with clear experimental evidence of elevated minimum inhibitory concentration (MIC). This has implications for molecular surveillance as sequences in clinical, agricultural, or environmental strains may differ in sequence from characterized & curated reference sequences. As such, the CARD team has been developing RGI\*metagenomics, a new tool using the BWT against both canonical CARD and predicted AMR resistance alleles from bulk resistome analyses (with use of the associated prevalence data to describe pathogen/plasmid distribution). In addition, we have been developing new k-mer based tools for pathogen-of-origin prediction from metagenomics data. Both of these software tools are unreleased beta-test versions, but we’ll try them on the gut microbiome data described above (this time as small subset of ~160,000 paired reads). Please note they do not yet incorporate AMR mutation screening (in development):

Return to your course account’s working directory, set up some more aliases, and prepare both the CARD data (downloaded earlier) and CARD Resistomes & Variants data for the BWT analysis:

alias card\_annotation="python3

/home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/card\_annotation.py"

alias wildcard\_annotation="python3

/home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/wildcard\_annotation.py"

alias card\_bowtie\_bwa="python3

/home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/card\_bowtie\_bwa.py"

mkdir card

cp card.json card/.

mkdir wildcard

cd wildcard

wget https://card.mcmaster.ca/latest/variants

tar -xvf variants

cd ..

ls

card\_annotation --input card/card.json

wildcard\_annotation --input\_directory wildcard --version 3.0.1

ls

cat card\_database\_v\*.fasta > reference.fasta

cat wildcard\_database\_v\*.fasta >> reference.fasta

We now have a merged CARD and CARD Variants reference FASTA file, plus associated metadata (in the card and wildcard directories). Take a look at the raw gut metagenomics data:

ls /home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/gut\_sample

less /home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/gut\_sample/gut\_R1.fastq

Ok, let’s perform the BWT for these data against the merged CARD reference data (BWT performed by Bowtie):

**DO NOT RUN THE COMMAND IN GRAY, WE WILL VIEW PRE-COMPILED RESULTS**

card\_bowtie\_bwa –h

card\_bowtie\_bwa

-1 /home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/gut\_sample/gut\_R1.fastq

-2 /home/ubuntu/CourseData/IDGE\_data/module4/repository/baits/gut\_sample/gut\_R2.fastq

-a bowtie2 -d reference.fasta -j card/card.json -i wildcard/index-for-model-sequences.txt

-n 4

cp /home/ubuntu/CourseData/IDGE\_data/module4/bowtie\_results/\* .

ls

Bowtie and other BWT files created a large index file of read mappings against the reference sequences in BAM format. We can use SAMTOOLS to look inside this file, using grep to limit the output to aminoglycoside phosphotransferase APH(6)-Id. Note the last three data columns are length of reference, # reads fully mapped, # reads mapped with unmapped flanking sequence:

samtools idxstats gut\_R1.sorted.bam | grep "APH(6)-Id"

The card\_bowtie\_bwa script take the BWT results and parses them relative to the CARD reference and variants data, producing metagenomics results indexed by allele or gene in the following files (the course GitHub repo contains EXCEL versions for us to view):

ls \*mapping\*

**Kmers and Pathogen-of-Origin for Metagenomics**

We can try one last beta CARD tool on these data, which is 66 bp kmers to predict pathogen-of-origin for these metagenomics data. We have pre-computed diagnostic kmers (for both pathogens and plasmids) from the CARD variants data and have a script that evaluates the BWT results for pathogen-of-origin. First we will use SAMTOOLS to extract the aligned sequences from the BAM file and split the results into a set of smaller files:

alias card\_pathogen\_id="python3

/home/ubuntu/CourseData/IDGE\_data/module4/repository/kmers/bwa\_widget\_pathid.py"

alias card\_kmer\_search="python3

/home/ubuntu/CourseData/IDGE\_data/module4/repository/kmers/metagenomics\_kmer\_query.py"

samtools view gut\_R1.sorted.bam | while read line; do awk '{print ">"$1"\_\_"$3"\_\_"$2"\n"$10}'; done

> gut\_R1.sorted.fasta

less gut\_R1.sorted.fasta

mkdir split-files

cd split-files

split -l 200000 -d --additional-suffix=.fasta ../gut\_R1.sorted.fasta gut\_R1.sorted

cd ..

Now we will search these putative AMR sequences from our metagenomics data for pre-compiled 66 bp kmers (the course GitHub repo contains EXCEL versions of the results for us to view):

card\_kmer\_search -h

card\_kmer\_search -i split-files -k 66

-s /home/ubuntu/CourseData/IDGE\_data/module4/kmer\_database/66k-species.json

-g /home/ubuntu/CourseData/IDGE\_data/module4/kmer\_database/66k-genus.json

-p /home/ubuntu/CourseData/IDGE\_data/module4/kmer\_database/66plasmid-kmers.txt

-b /home/ubuntu/CourseData/IDGE\_data/module4/kmer\_database/66both-kmers.txt

-c /home/ubuntu/CourseData/IDGE\_data/module4/kmer\_database/66chr-kmers.txt

-o gut\_R1

card\_pathogen\_id –h

card\_pathogen\_id -j gut\_R1-66k.json -at gut\_R1.allele\_mapping\_data.txt

-gt gut\_R1.gene\_mapping\_data.txt -o gut\_R1-66k