From Detection to Amplification: Metagenomics Perspectives on Antimicrobial Resistance

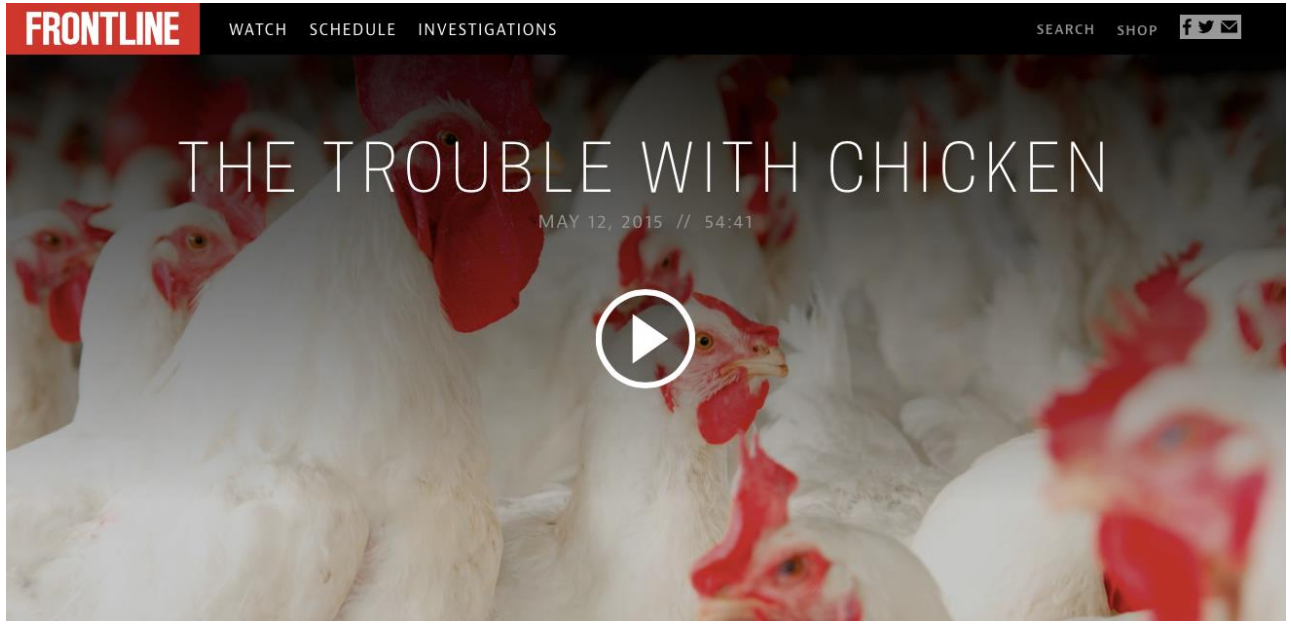
Christina Boucher

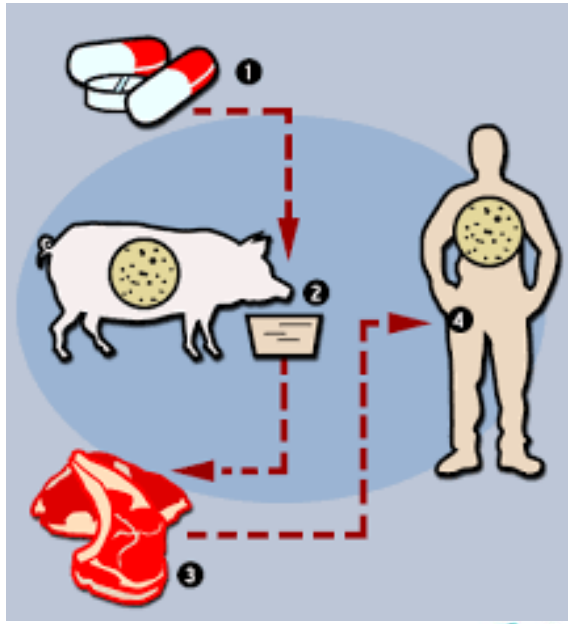
Dept of Computer and Information Science and Engineering

Herbert Wertheim College of Engineering

www.christinaboucher.com

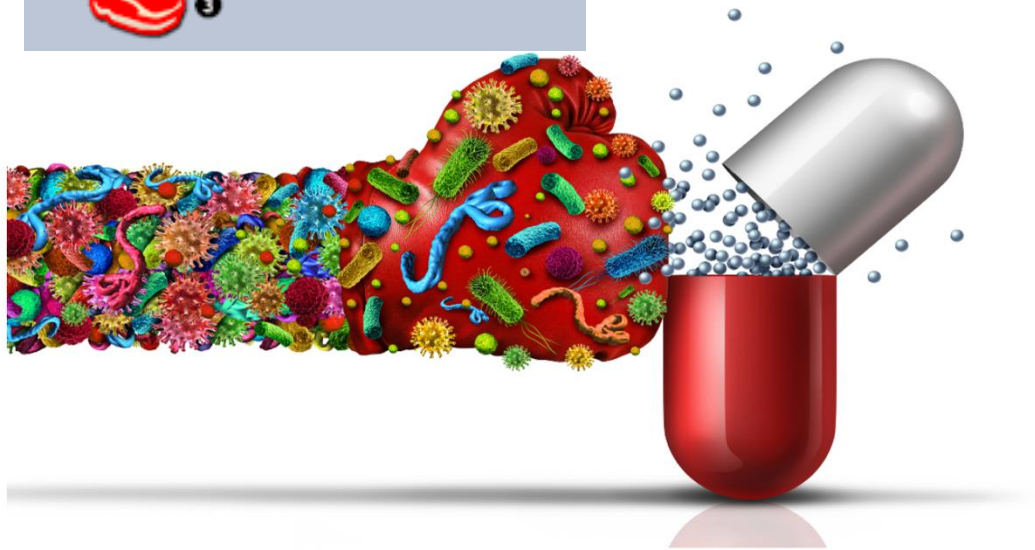
UF | UNIVERSITY of FLORIDA





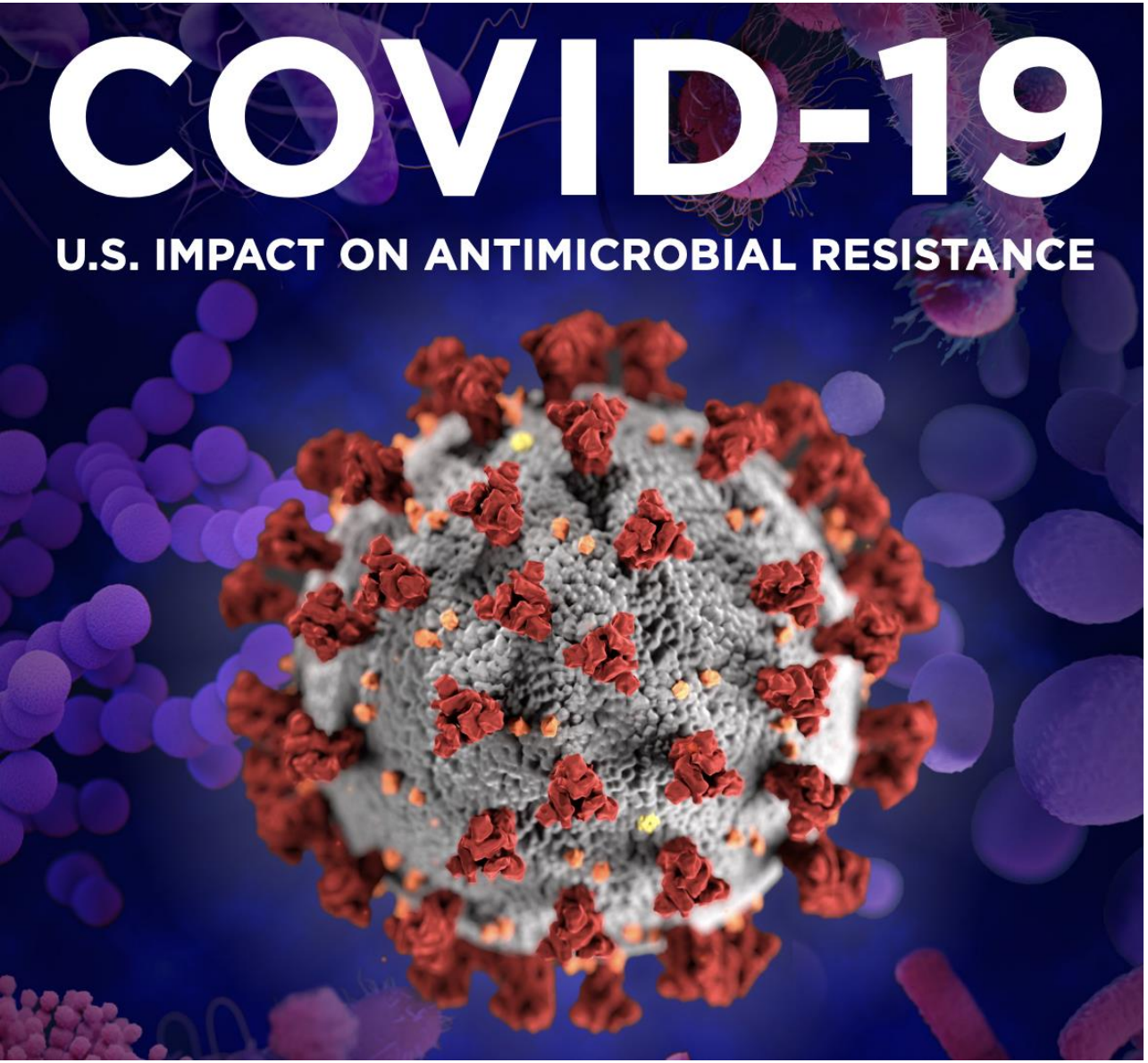
04-27-2022

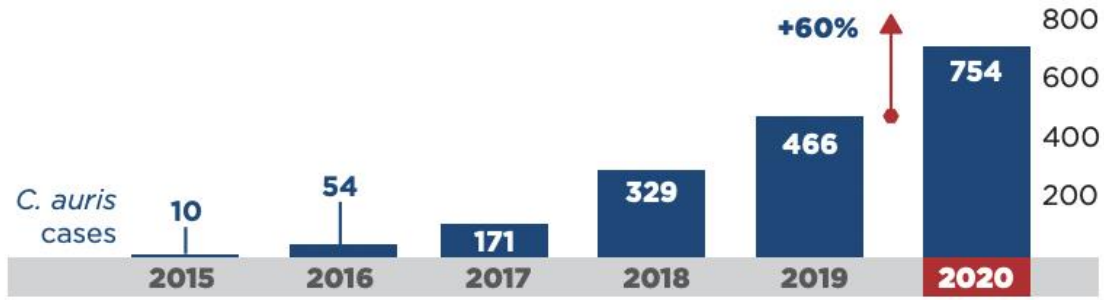
Pigs and humans are sharing antibiotic-resistance genes



EU figures show high levels of antibiotic resistance in foodborne bacteria

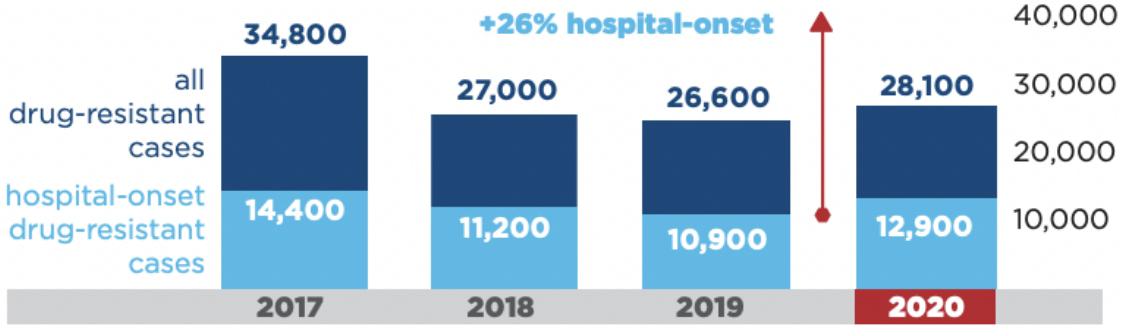
By Joe Whitworth on April 6, 2022



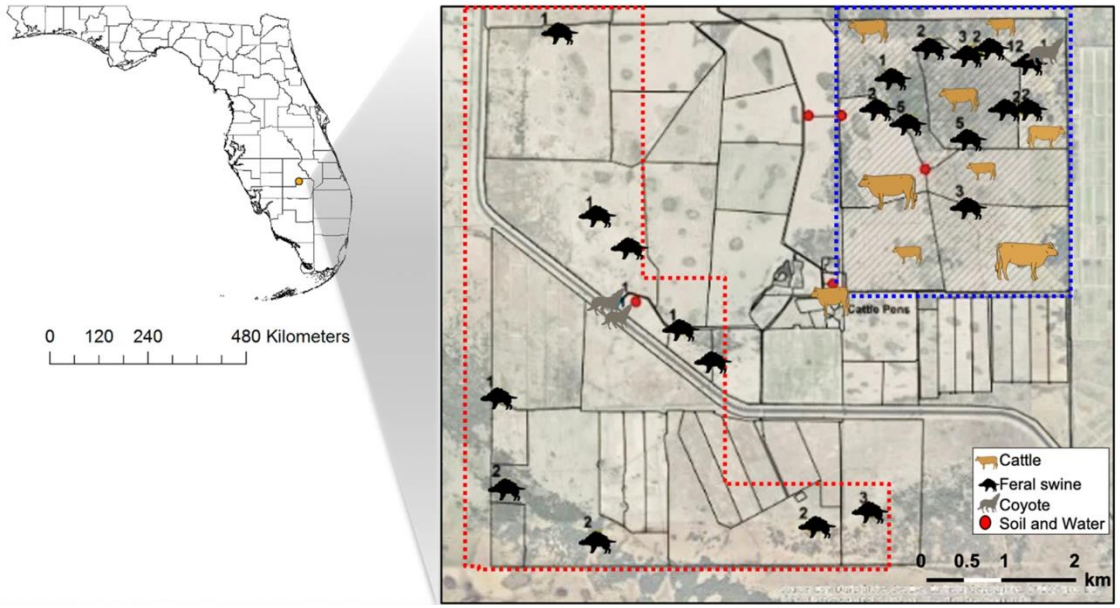


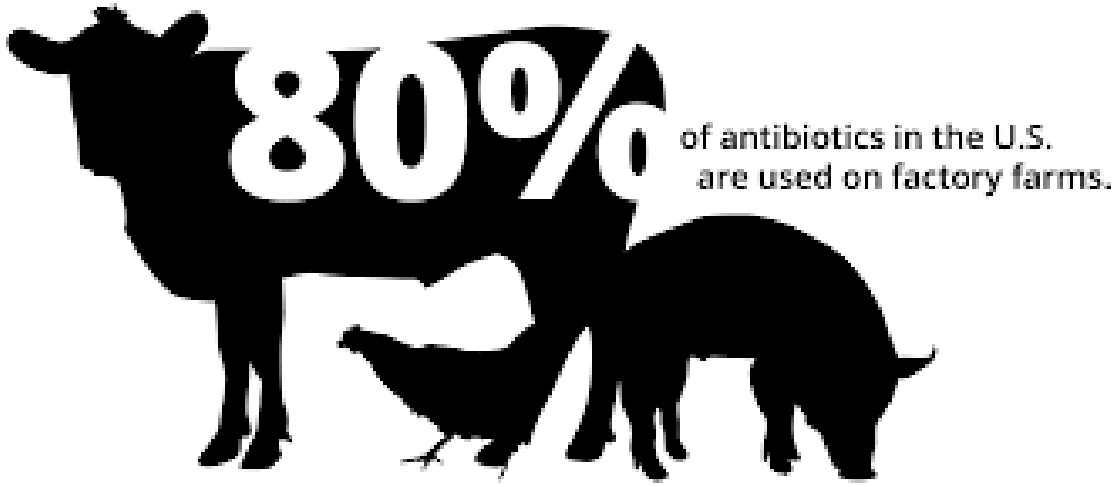
COVID-19: U.S. Impact on Antimicrobial Resistance, Special Report 2022

Drug-resistant *Candida* cases decreased until 2020, when hospital-onset cases increased during the COVID-19 pandemic.

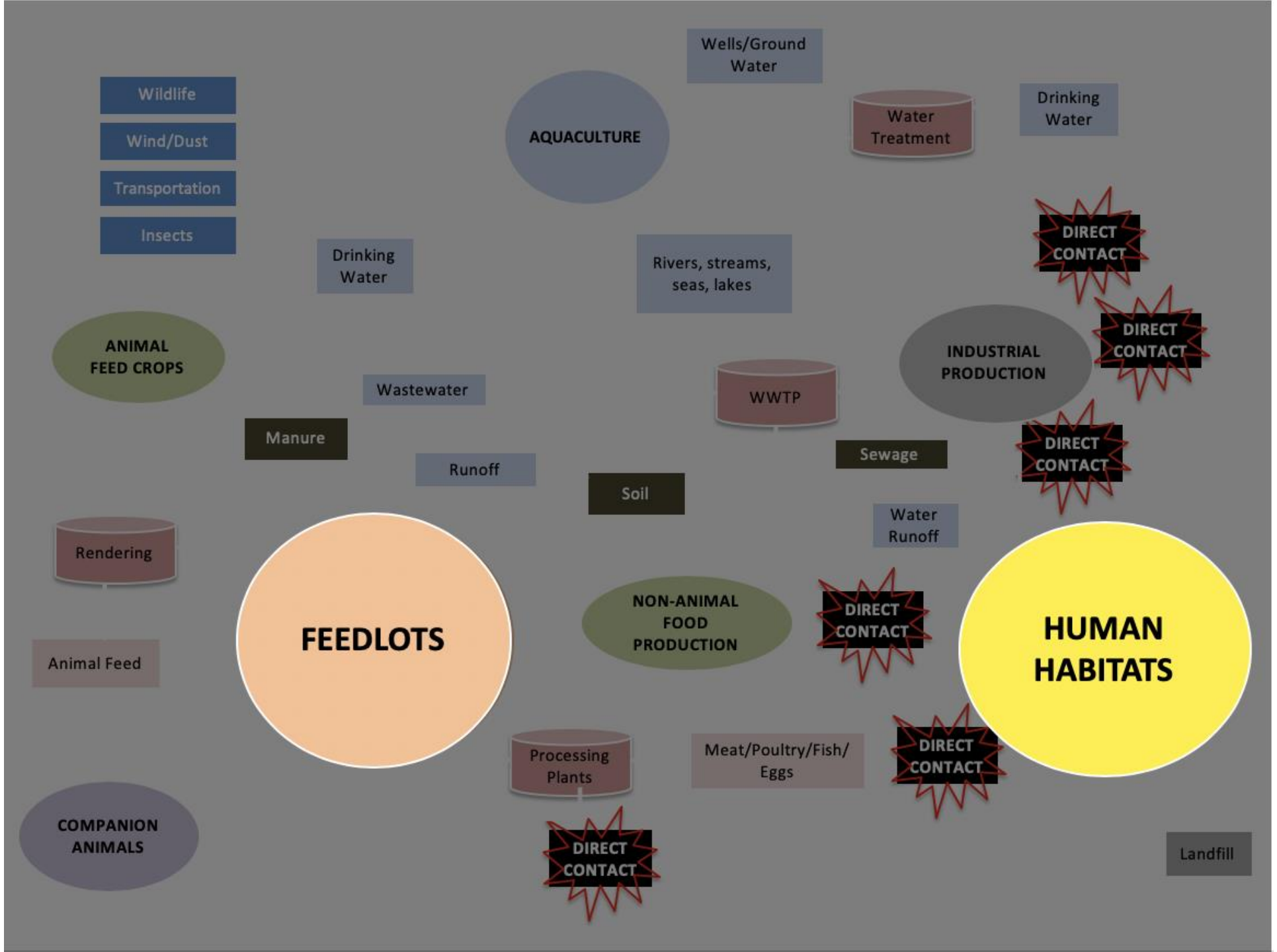


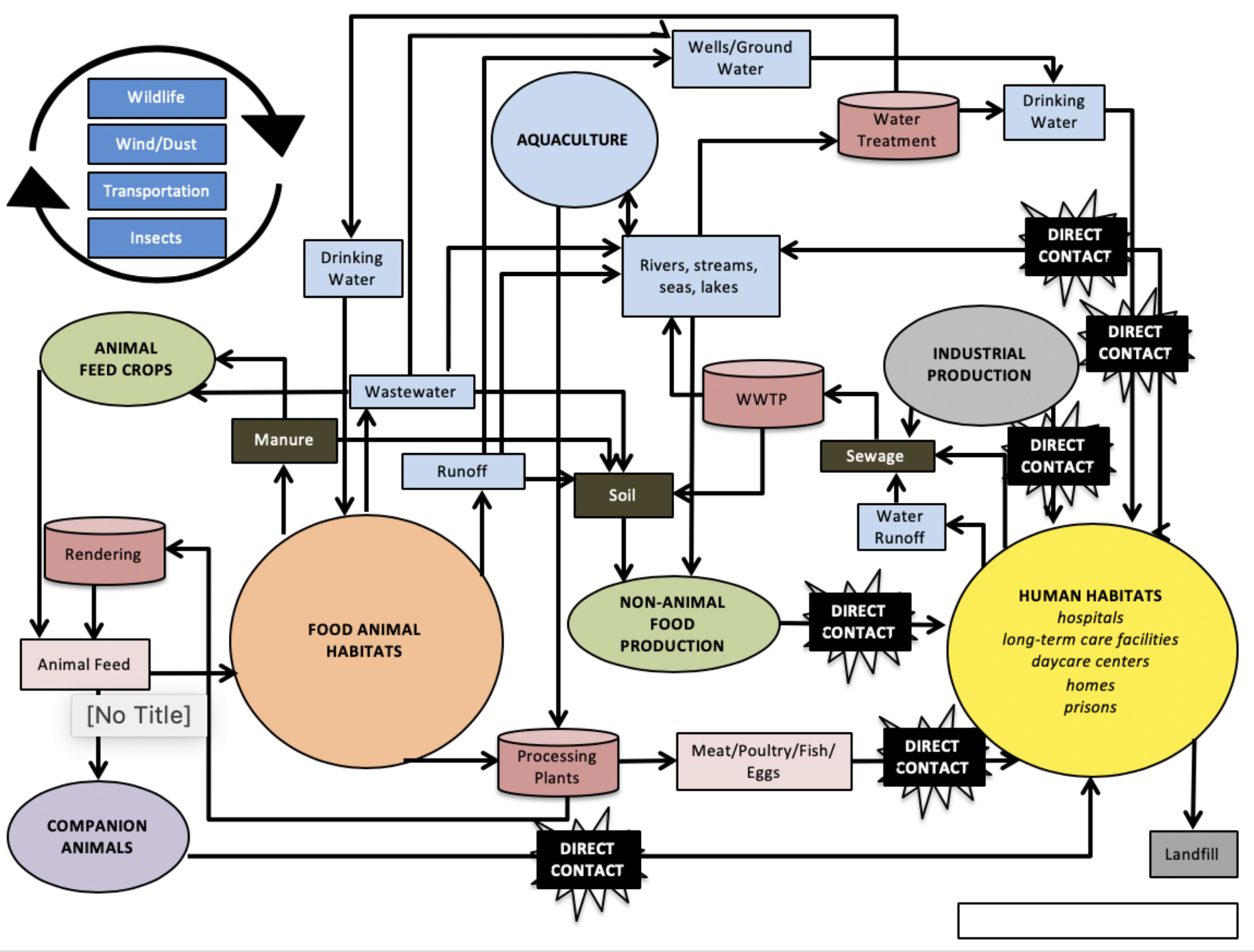
AR Pathogens Cause Infections Across the One Health Spectrum





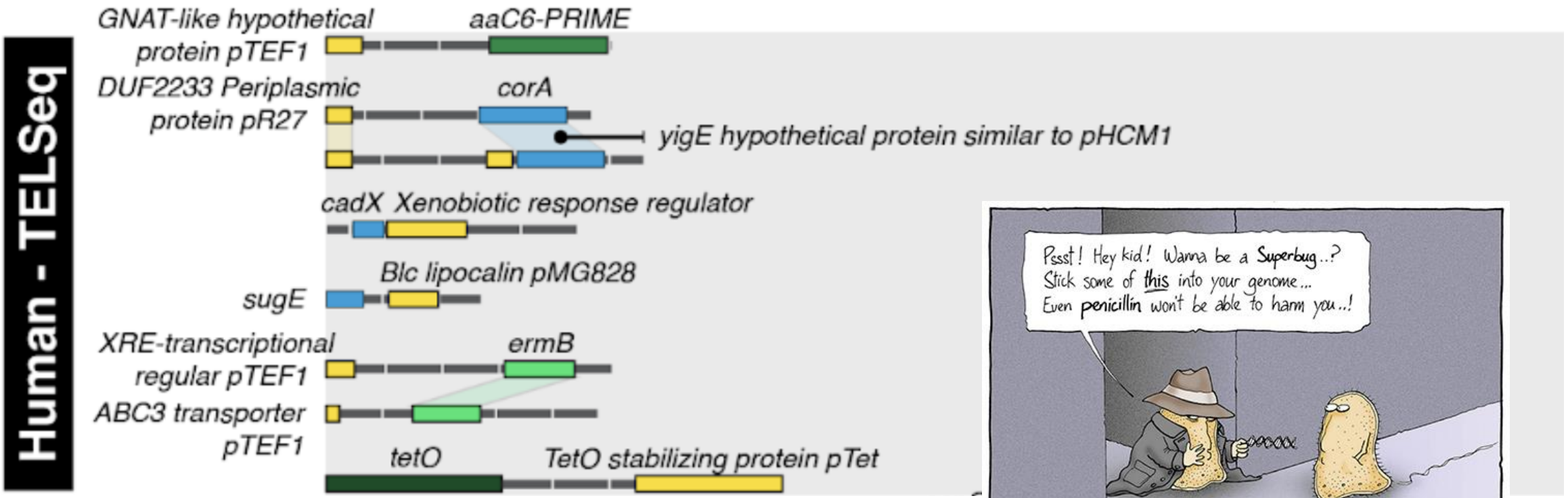
[Transmission of antibiotic resistance at the wildlife-livestock interface](#) S Lee et al. Communications biology 5 (1), 1-12


Wells/Ground Water
Wildlife
Wind/Dust
Transportation
Insects
AQUACULTURE
Water Treatment
Drinking Water
DIRECT CONTACT
Drinking Water
Rivers, streams, seas, lakes
DIRECT CONTACT
ANIMAL FEED CROPS
INDUSTRIAL PRODUCTION
Wastewater
WWTP
Manure
DIRECT CONTACT
Sewage
Runoff
Soil
Water Runoff
Rendering
FEEDLOTS
NON-ANIMAL FOOD PRODUCTION
DIRECT CONTACT
HUMAN HABITATS
Animal Feed
Processing Plants
Meat/Poultry/Fish/ Eggs
DIRECT CONTACT
COMPANION ANIMALS
DIRECT CONTACT
Landfill


Wildlife
Wind/Dust
Transportation
Insects
Wells/Ground Water
AQUACULTURE
Water Treatment
Drinking Water
Drinking Water
Rivers, streams, seas, lakes
DIRECT CONTACT
DIRECT CONTACT
ANIMAL FEED CROPS
INDUSTRIAL PRODUCTION
Wastewater
WWTP
Manure
Runoff
Soil
Sewage
DIRECT CONTACT
Water Runoff
Rendering
FOOD ANIMAL HABITATS
NON-ANIMAL FOOD PRODUCTION
DIRECT CONTACT
HUMAN HABITATS
hospitals
long-term care facilities
daycare centers
homes
prisons
Animal Feed
[No Title]
Processing Plants
Meat/Poultry/Fish/ Eggs
DIRECT CONTACT
COMPANION ANIMALS
DIRECT CONTACT
Landfill

Detection of Resistance

Antimicrobial Resistant Genes (ARGs)



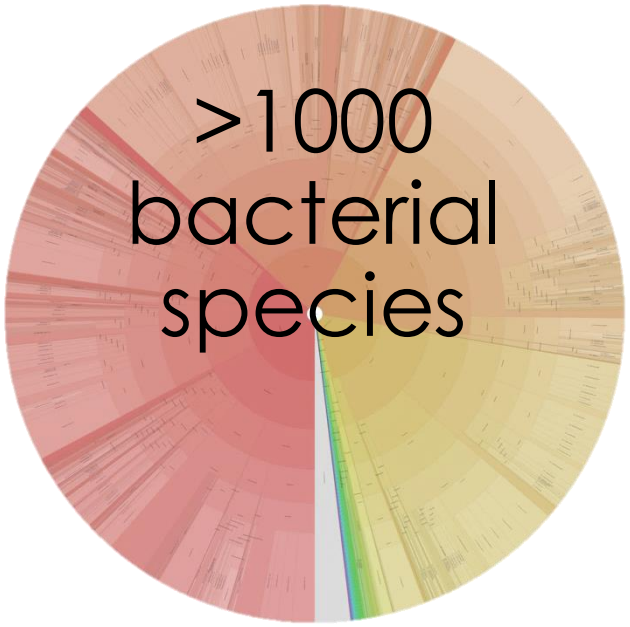
It was on a short-cut through the hospital kitchens that Albert was first approached by a member of the Antibiotic Resistance.

Culture Based Methods

Old Method

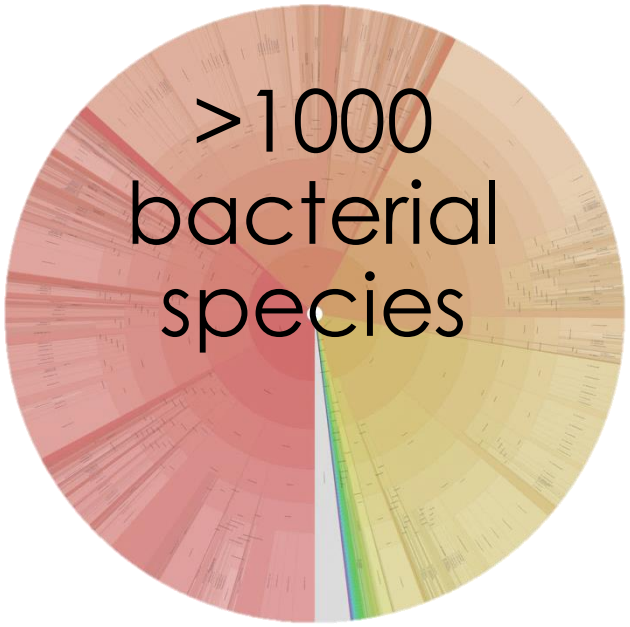
Microbiome is a Complex System

Old Method vs. **Reality**

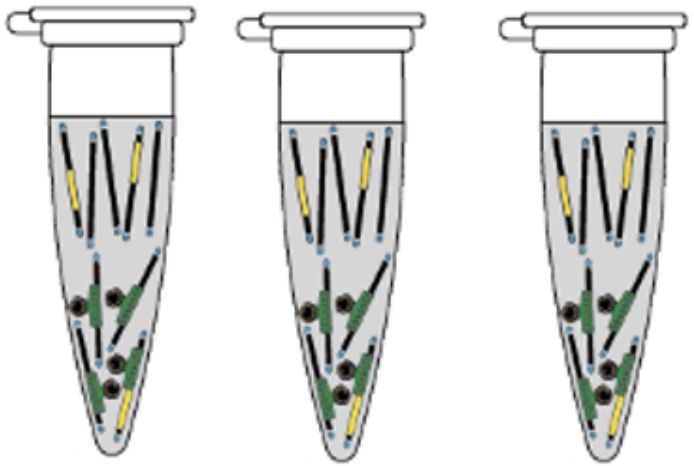


Microbiome is a Complex System

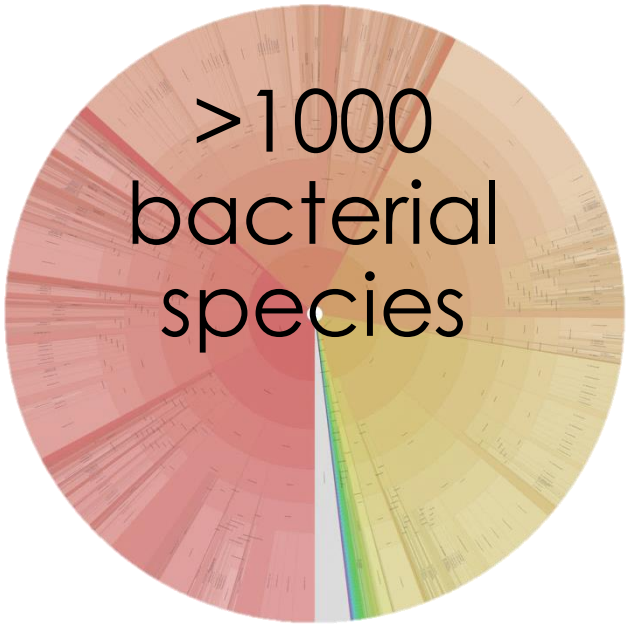
Old Method vs. **Reality**



Microbiome is a Complex System



Reality



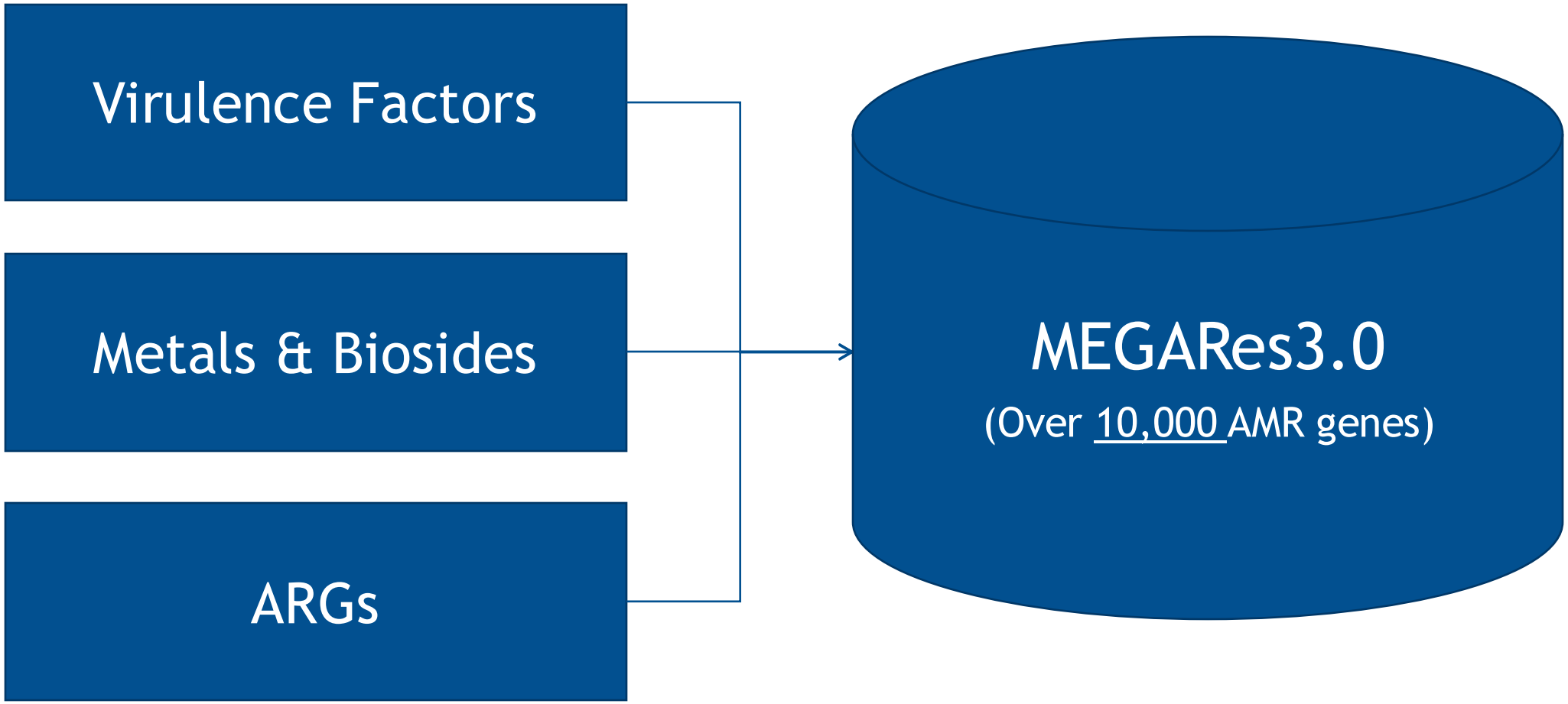
Introduction of the Resistome

Resistome:

All ARGs in both pathogenic and non-pathogenic bacteria

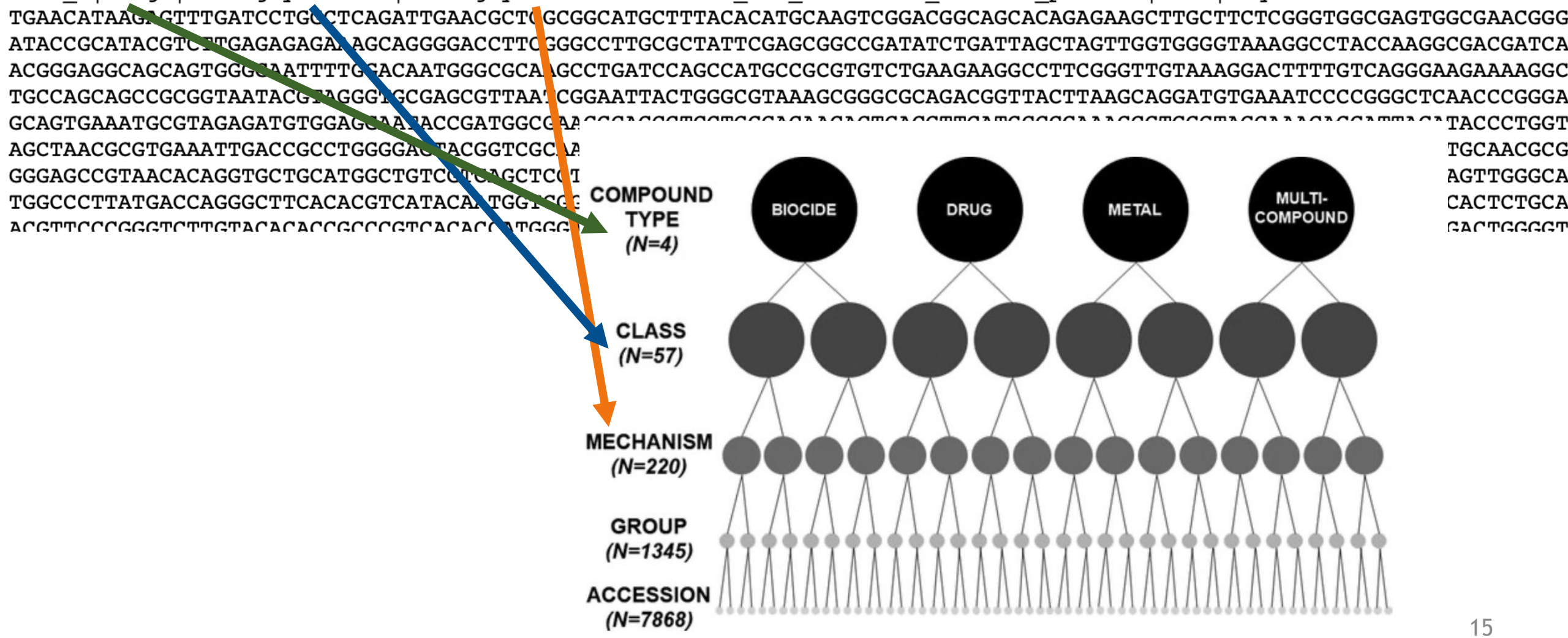
>1000
bacterial species

Introduction of MEGARes



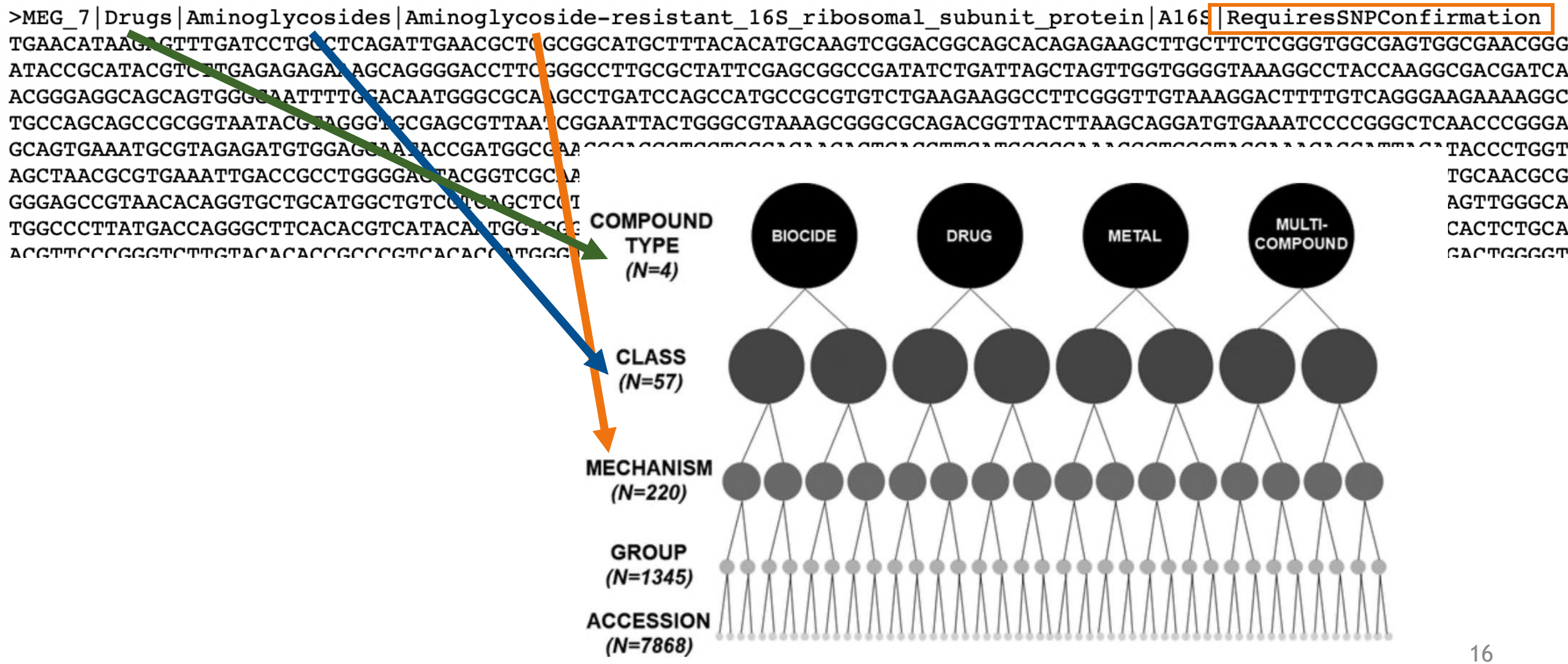
Introduction of MEGARes

>MEG_7|Drugs|Aminoglycosides|Aminoglycoside-resistant_16S_ribosomal_subunit_protein|A16S|RequiresSNPConfirmation

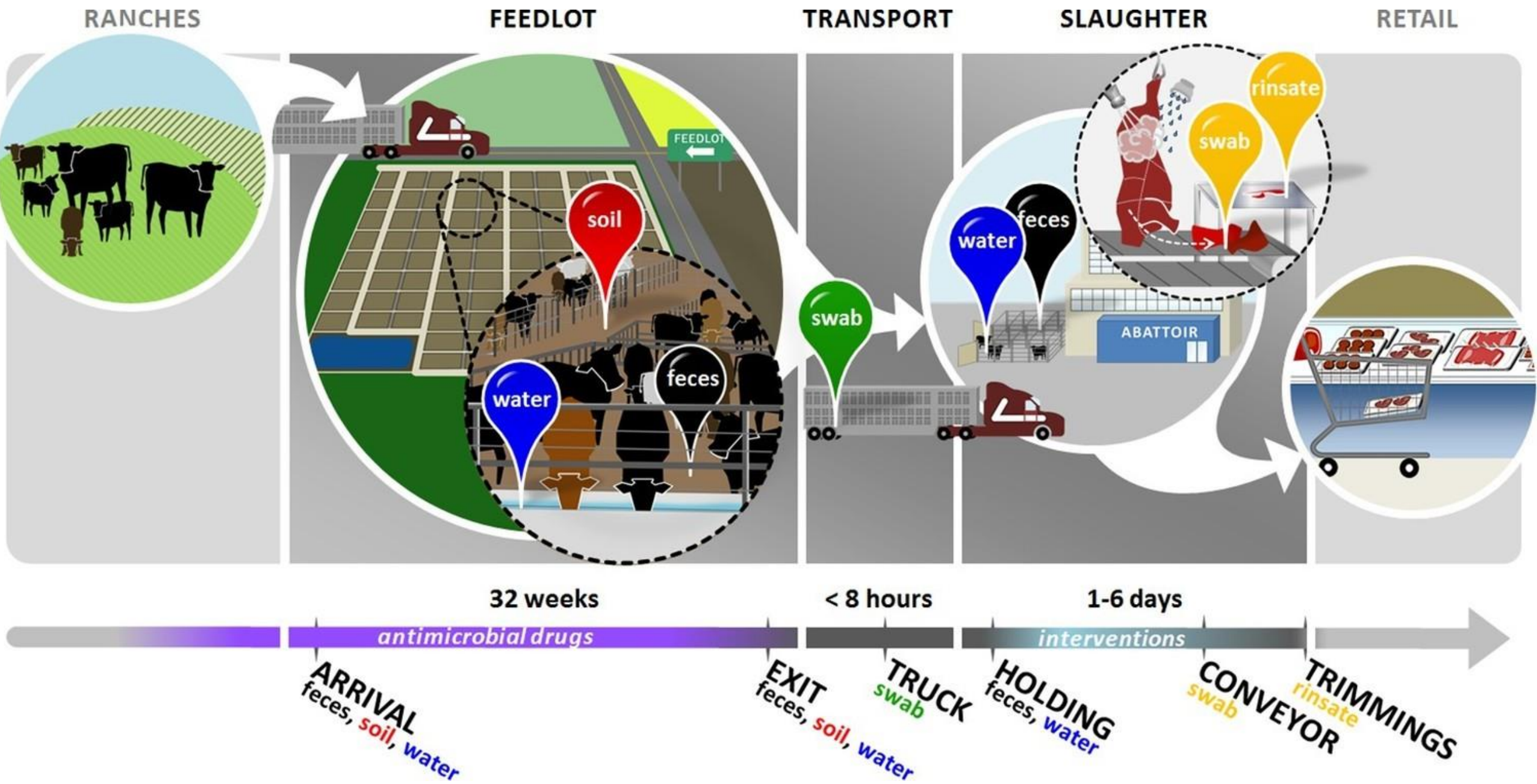


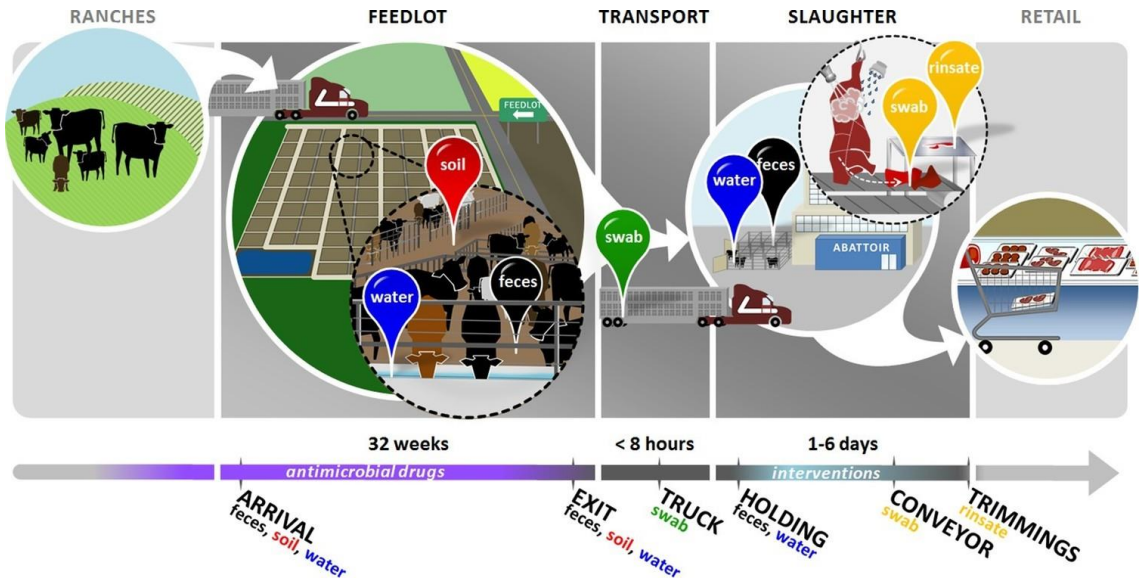
Introduction of MEGARes

>MEG_7|Drugs|Aminoglycosides|Aminoglycoside-resistant_16S_ribosomal_subunit_protein|A16S|RequiresSNPConfirmation



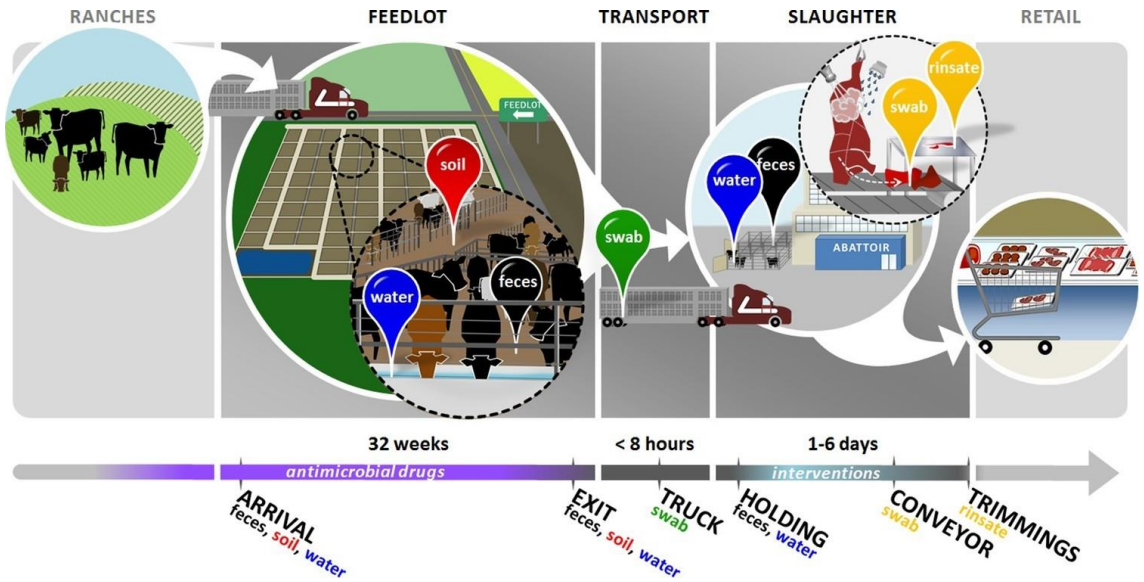
A Case Study of Resistance


RANCHES
FEEDLOT
TRANSPORT
SLAUGHTER
RETAIL
FEEDLOT
soil
water
feces
swab
water
feces
rinsate
swab
ABATTOIR
32 weeks
< 8 hours
1-6 days
antimicrobial drugs
interventions
ARRIVAL
feces, soil, water
EXIT
feces, soil, water
TRUCK
swab
HOLDING
feces, water
CONVEYOR
swab
TRIMMINGS
rinsate



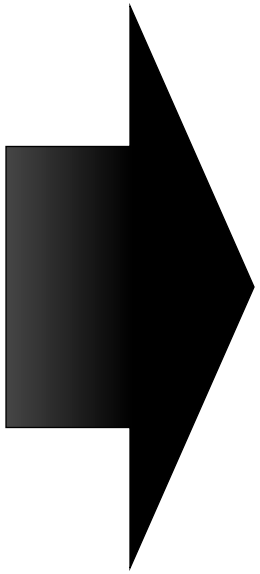
Sample Preparation

Sequencing



Sample Preparation

Sequencing



INPUT FASTQ Data

Quality Control & Host Filtering

Microbiome Analysis

Build krakenDB from RefSeq

k-mer match to krakenDB

Taxonomic naming

OUTPUT Microbiome

Resistome Analysis

Alignment to MEGaRES

Resistome Analyzer

Rarefaction Analyzer

Count Deduplication

OUTPUT Resistome

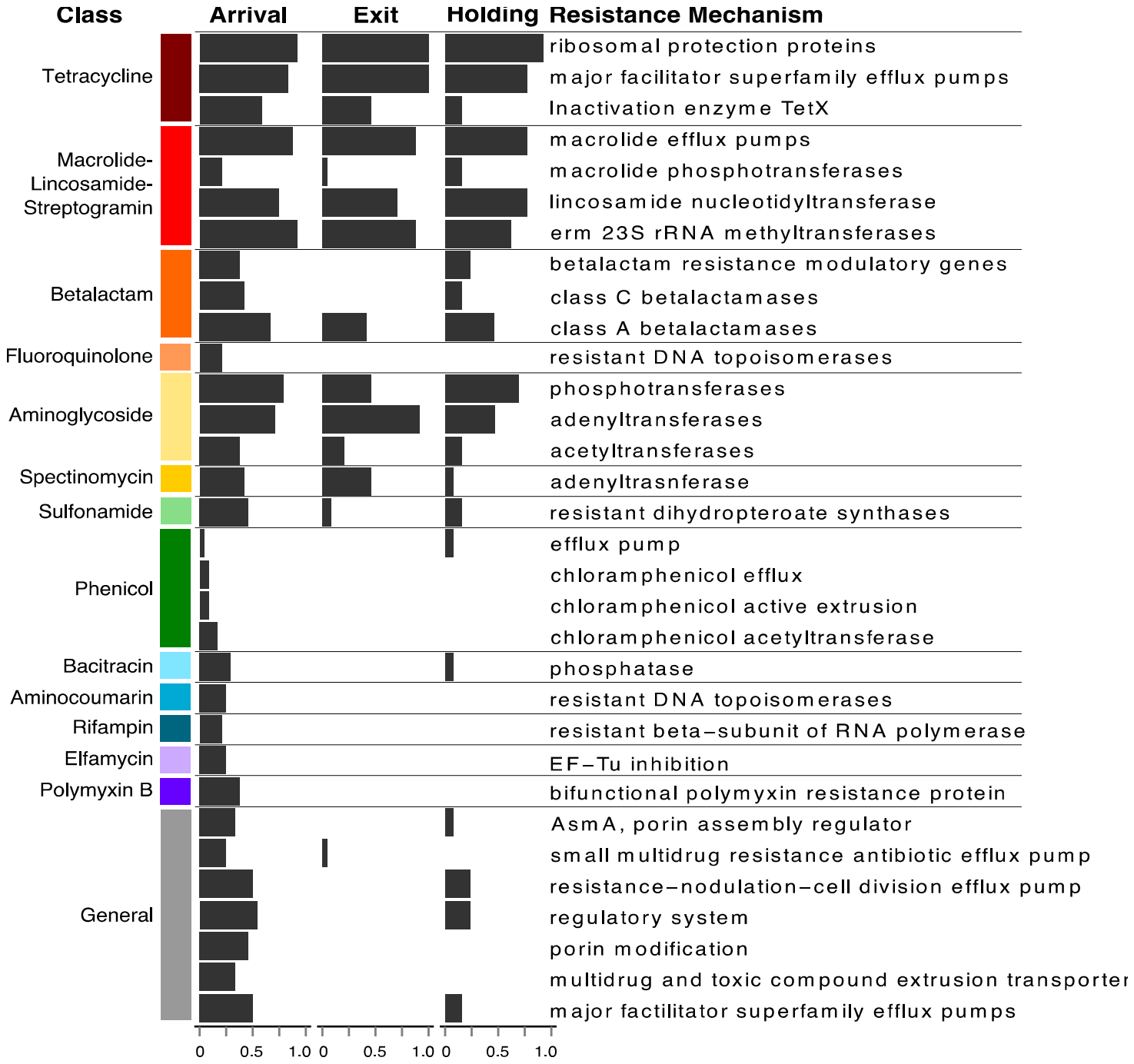
OUTPUT Rarefaction

Extract "RequiresSNPConfirmation"

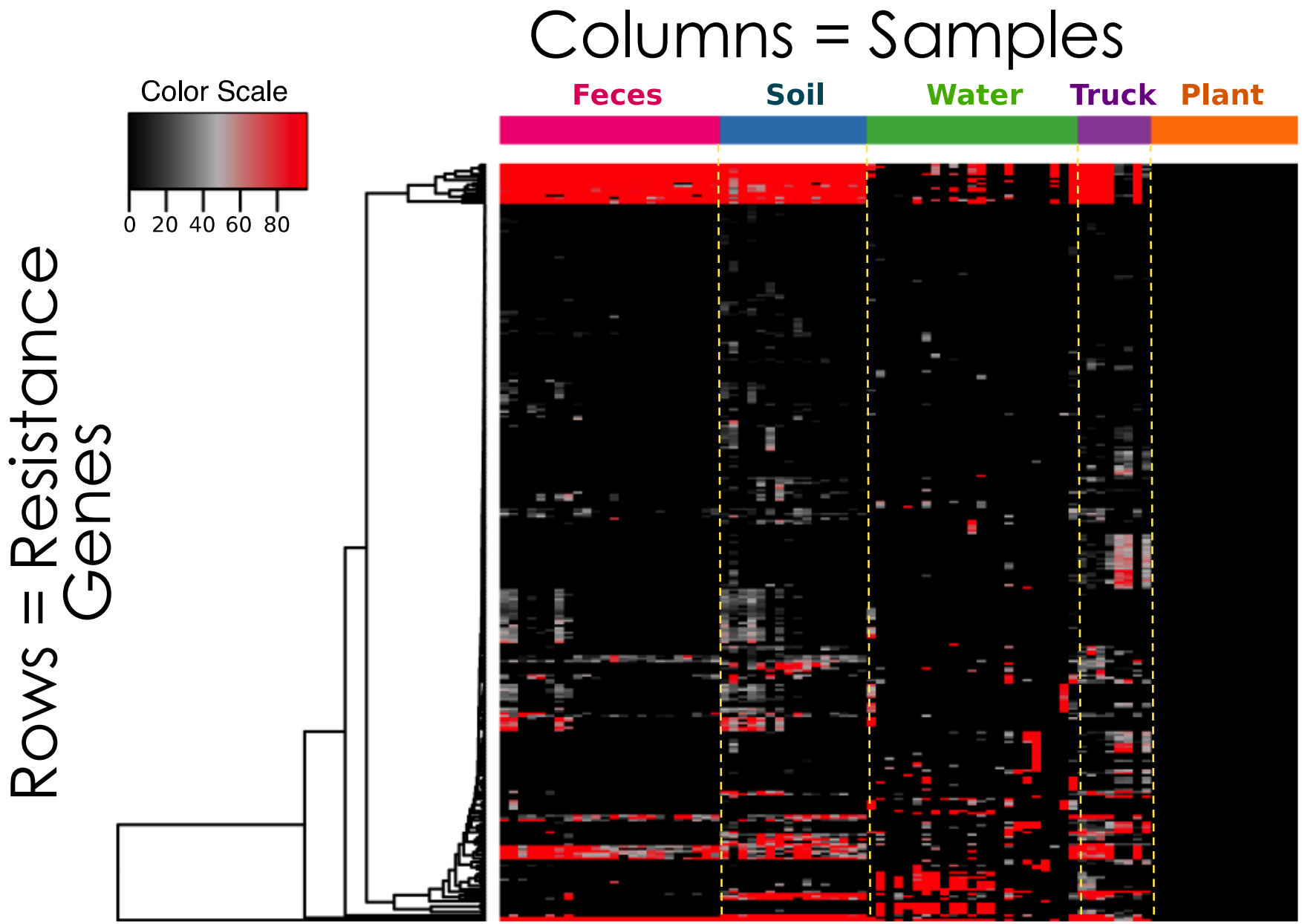
Resistance Gene Identifier

OUTPUT "SNP-confirmed" Resistome

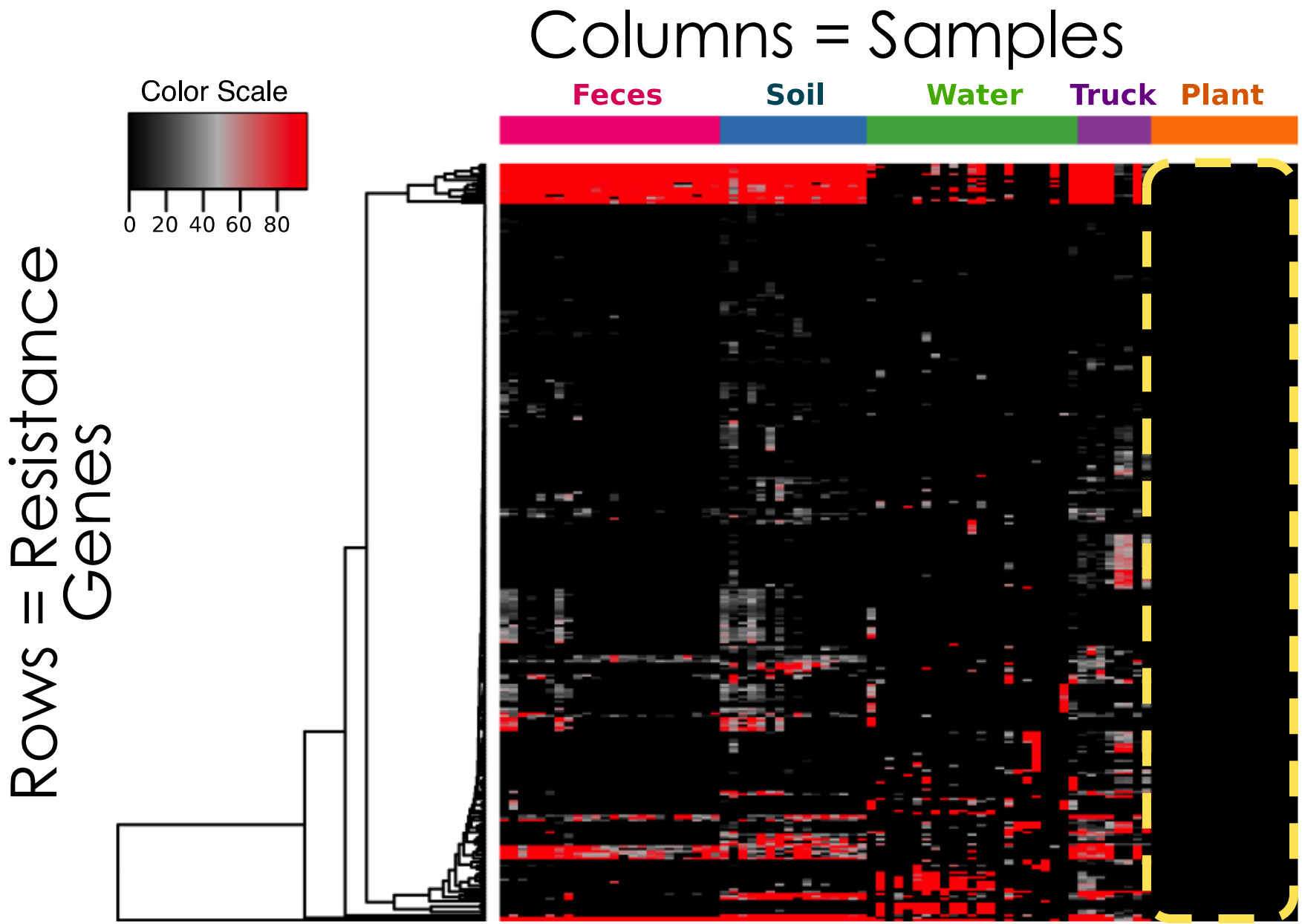
AMR++ Pipeline


Class
Arrival
Exit
Holding
Resistance Mechanism
Tetracycline
ribosomal protection proteins
major facilitator superfamily efflux pumps
Inactivation enzyme TetX
Macrolide-Lincosamide-Streptogramin
macrolide efflux pumps
macrolide phosphotransferases
lincosamide nucleotidyltransferase
erm 23S rRNA methyltransferases
Betalactam
betalactam resistance modulatory genes
class C betalactamases
class A betalactamases
Fluoroquinolone
resistant DNA topoisomerases
Aminoglycoside
phosphotransferases
adenyltransferases
acetyltransferases
Spectinomycin
adenyltrasnferase
Sulfonamide
resistant dihydropteroate synthases
Phenicol
efflux pump
chloramphenicol efflux
chloramphenicol active extrusion
chloramphenicol acetyltransferase
Bacitracin
phosphatase
Aminocoumarin
resistant DNA topoisomerases
Rifampin
resistant beta−subunit of RNA polymerase
Elfamycin
EF−Tu inhibition
Polymyxin B
bifunctional polymyxin resistance protein
General
AsmA, porin assembly regulator
small multidrug resistance antibiotic efflux pump
resistance−nodulation−cell division efflux pump
regulatory system
porin modification
multidrug and toxic compound extrusion transporter
major factilitator superfamily efflux pumps
0
0.5
1.0
0
0.5
1.0
0
0.5
1.0

The resistome clustered by matrix

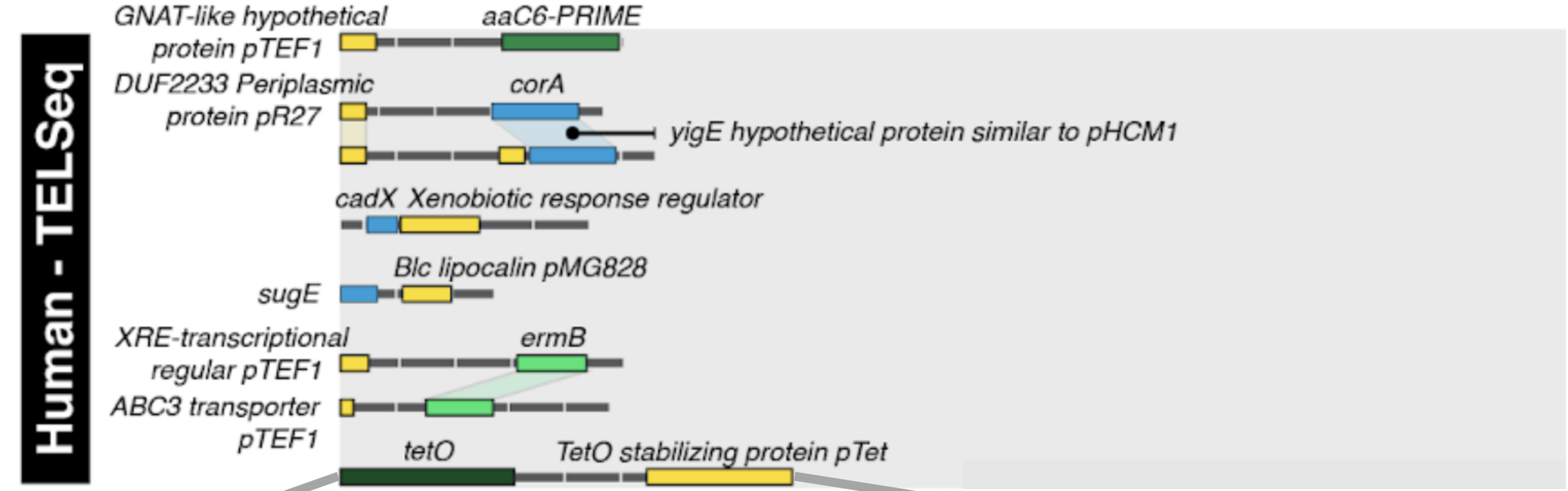


No ARDs identified in plant samples



Amplification of Resistance

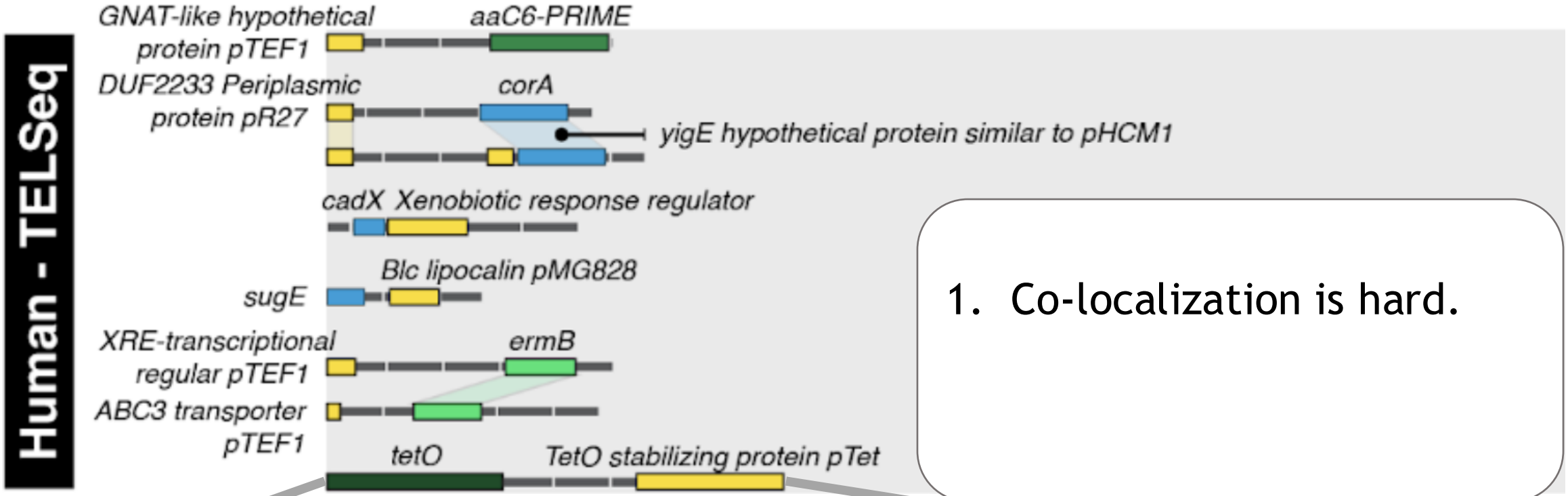
Challenges Related to Metagenomics



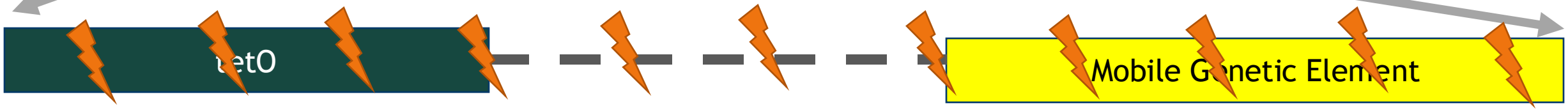
tetO

Mobile Genetic Element

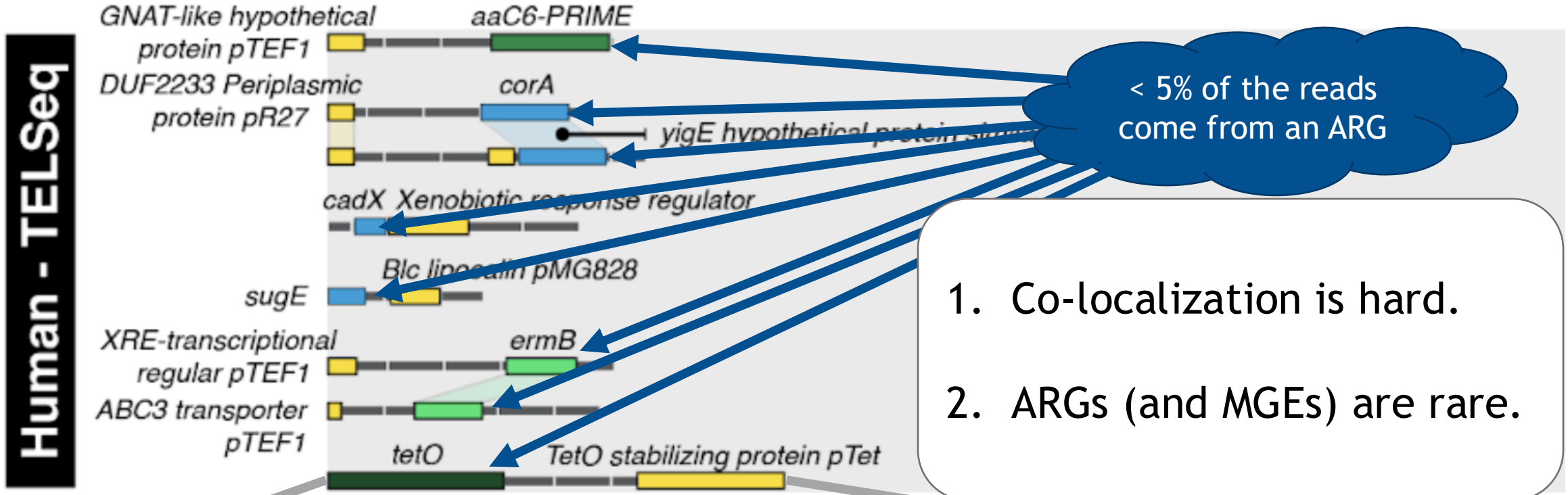
Challenges Related to Metagenomics



1. Co-localization is hard.

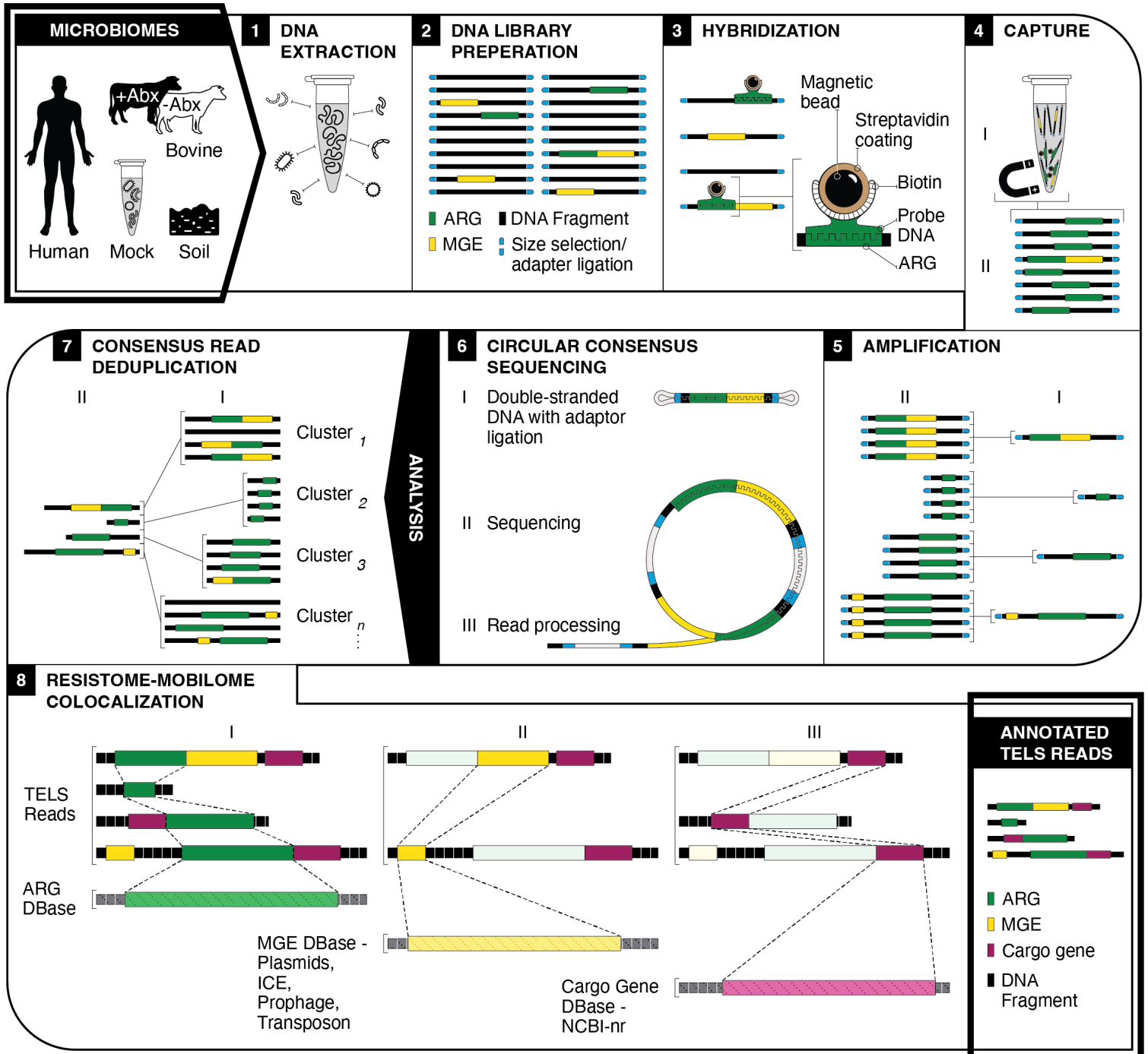


Challenges Related to Metagenomics

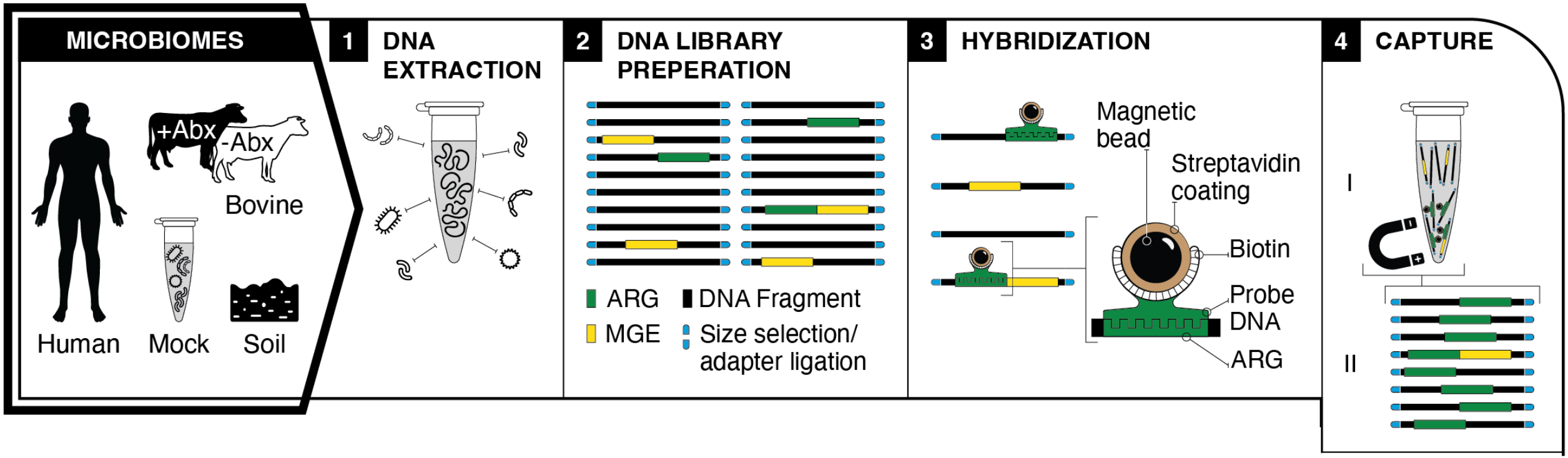


Presequencing

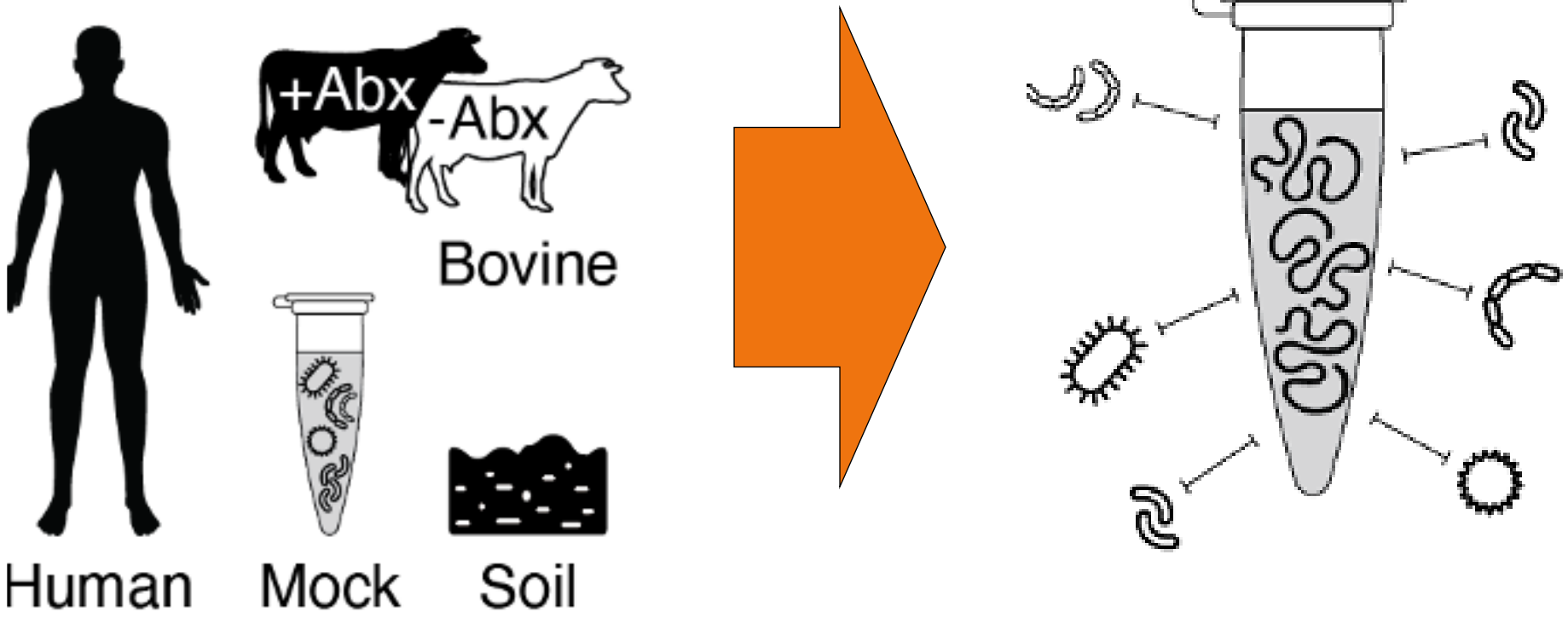
Postsequencing



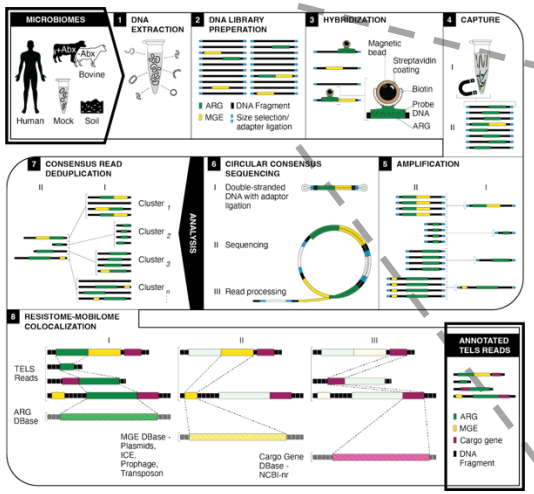
Introduction to Amplification

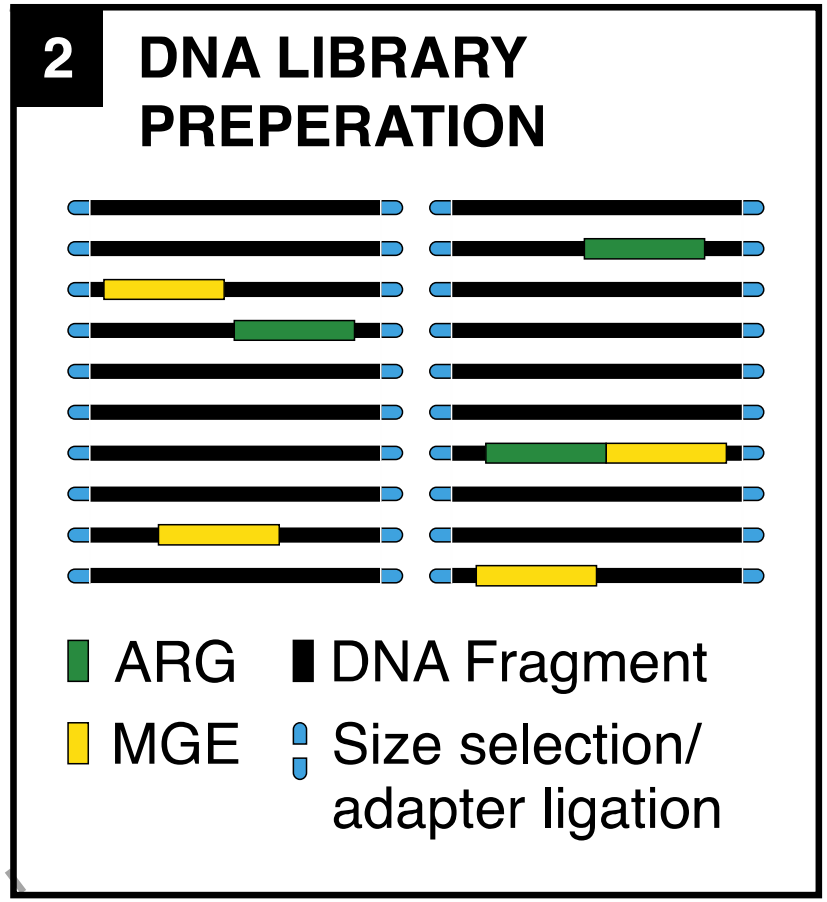


Step 1: DNA Extraction

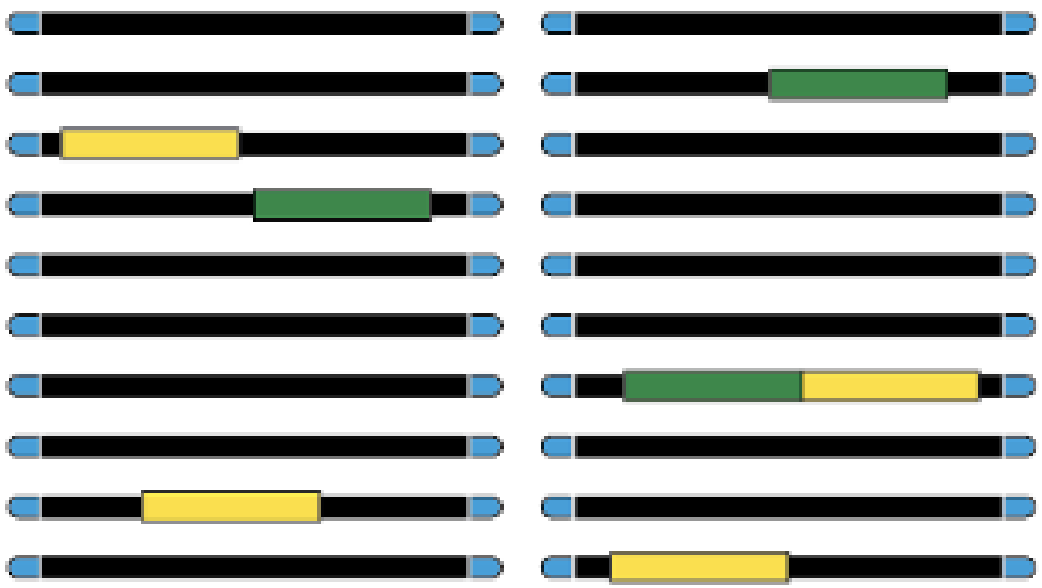


Application of Baits



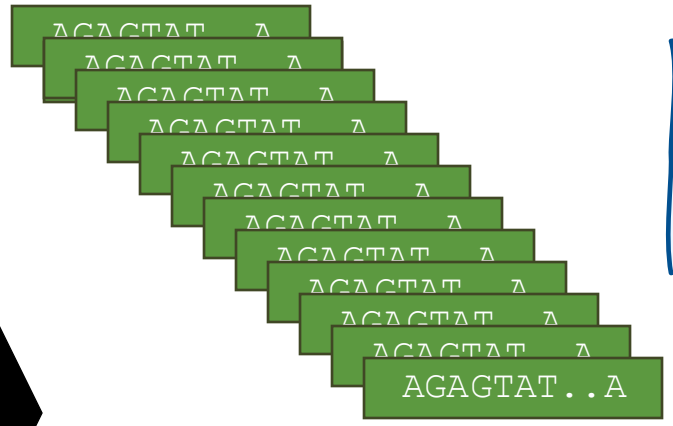


Amplification via Probes

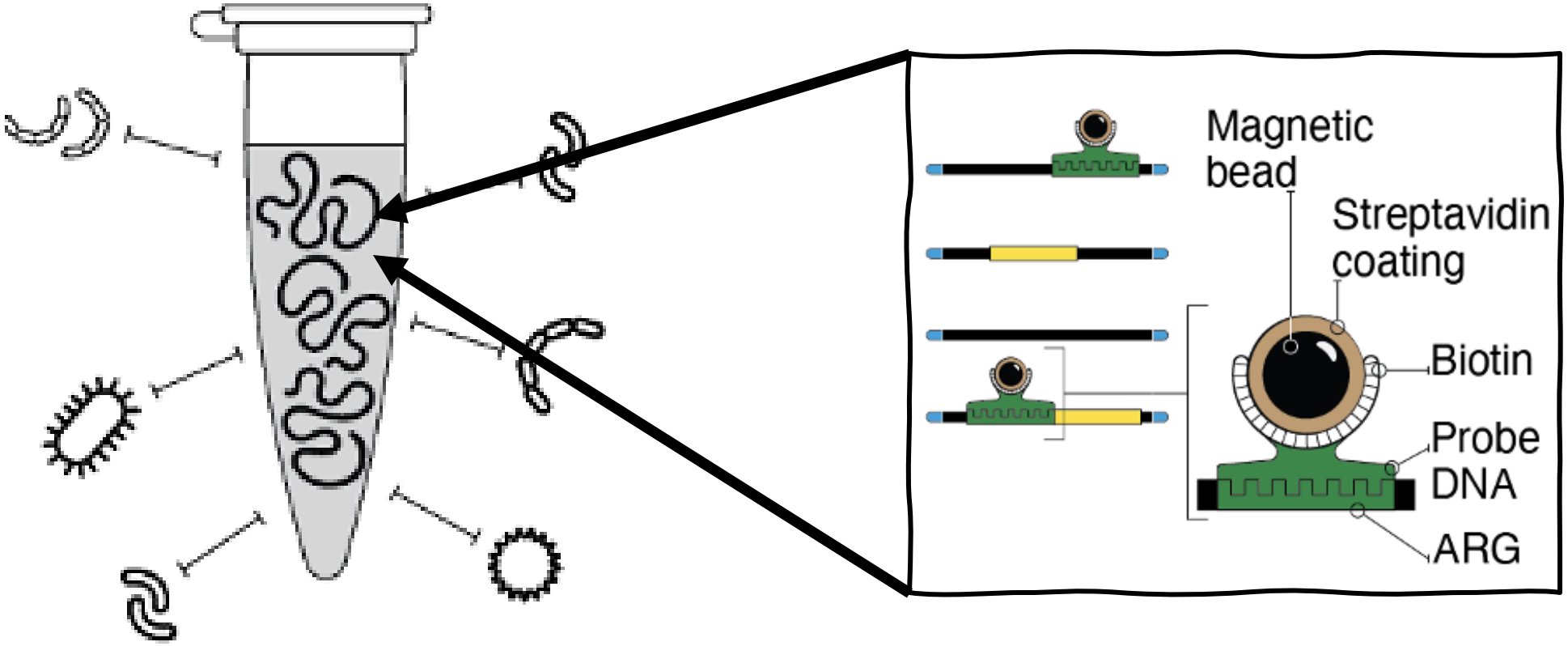


ARG DNA Fragment

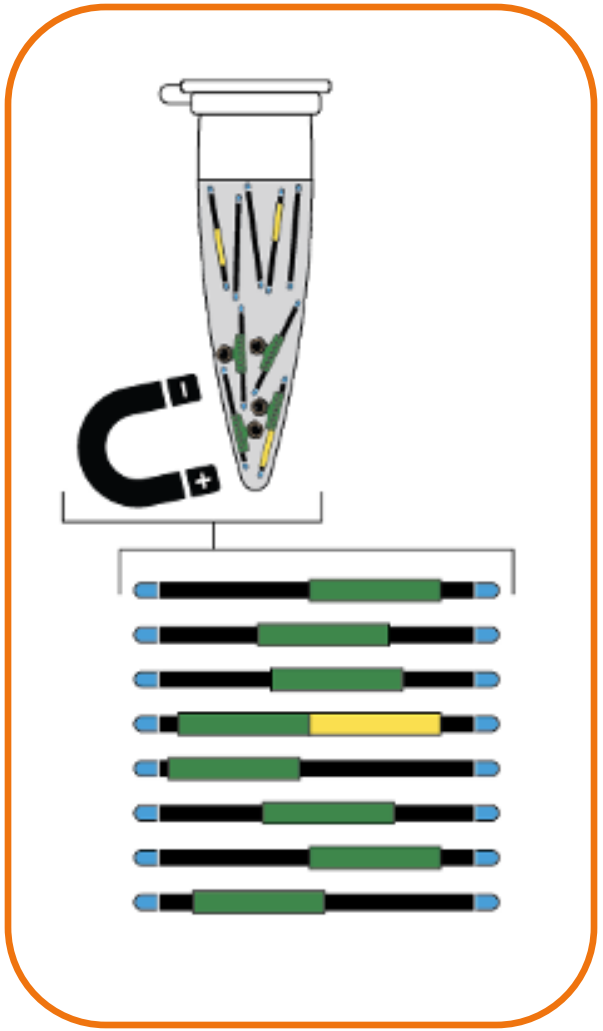
MGE Size selection/ adapter ligation

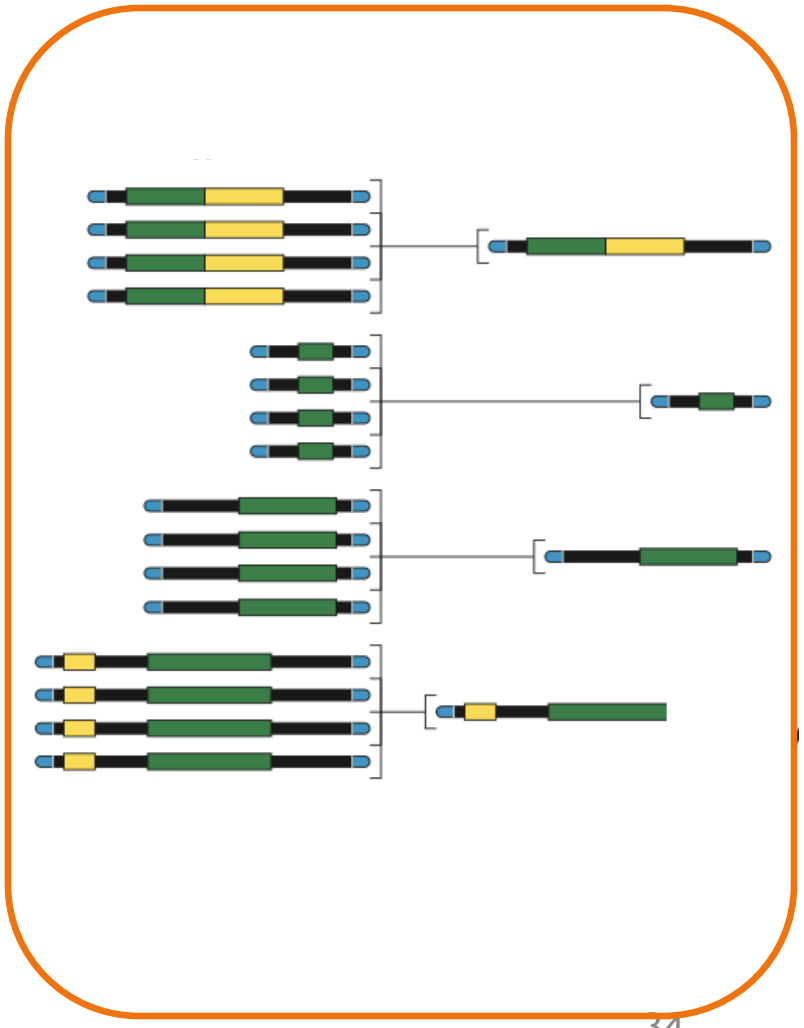


Amplification via Probes

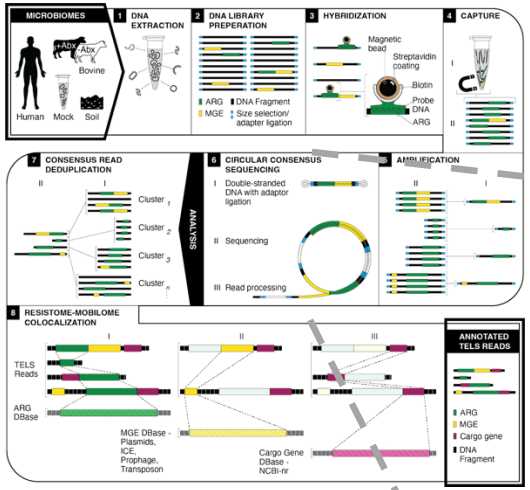


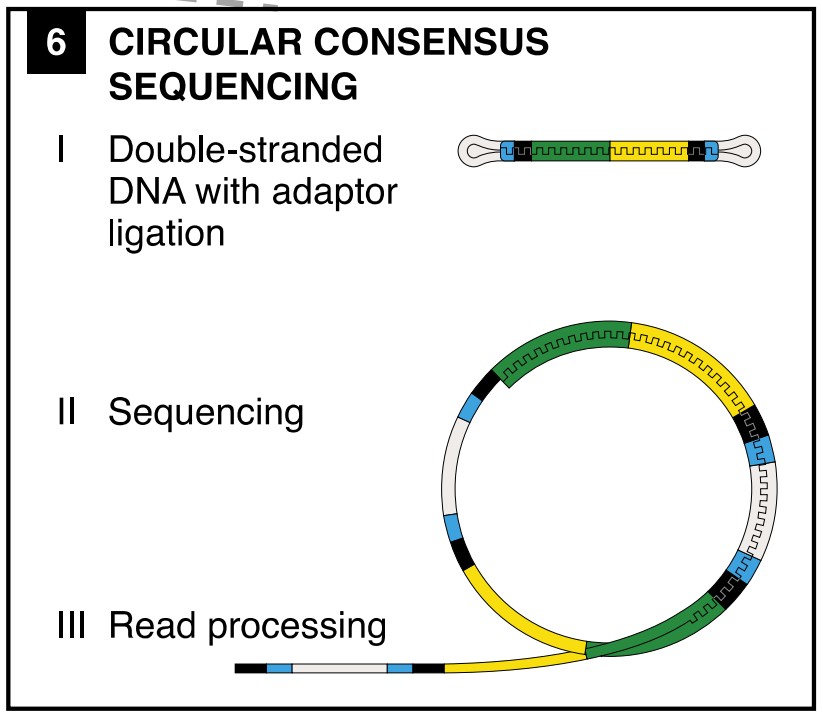
4: Sequencing

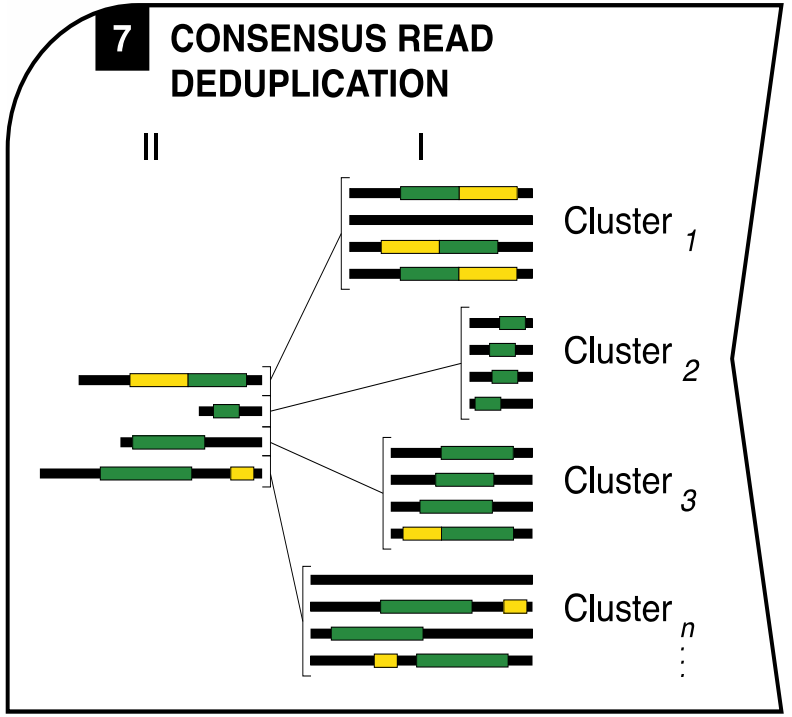




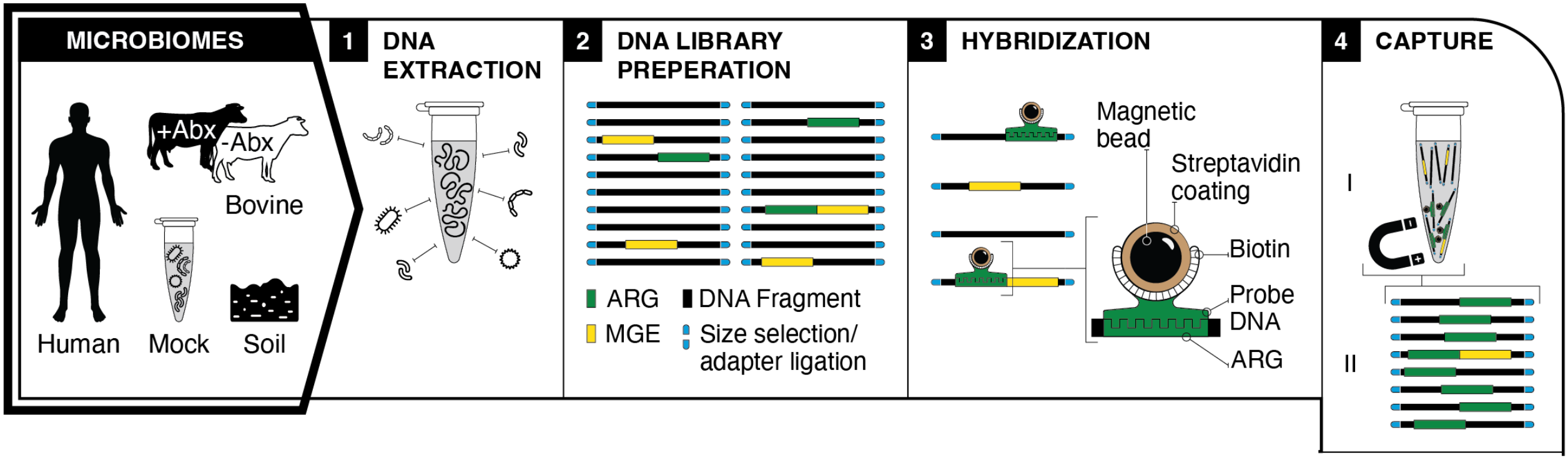
Pre-analysis



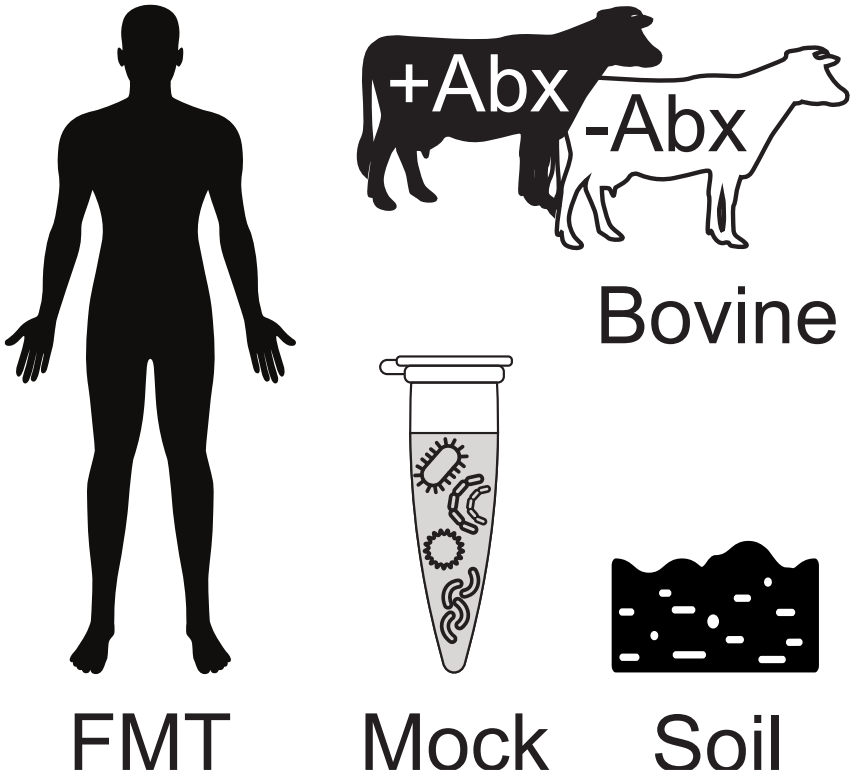




Introduction to Amplification

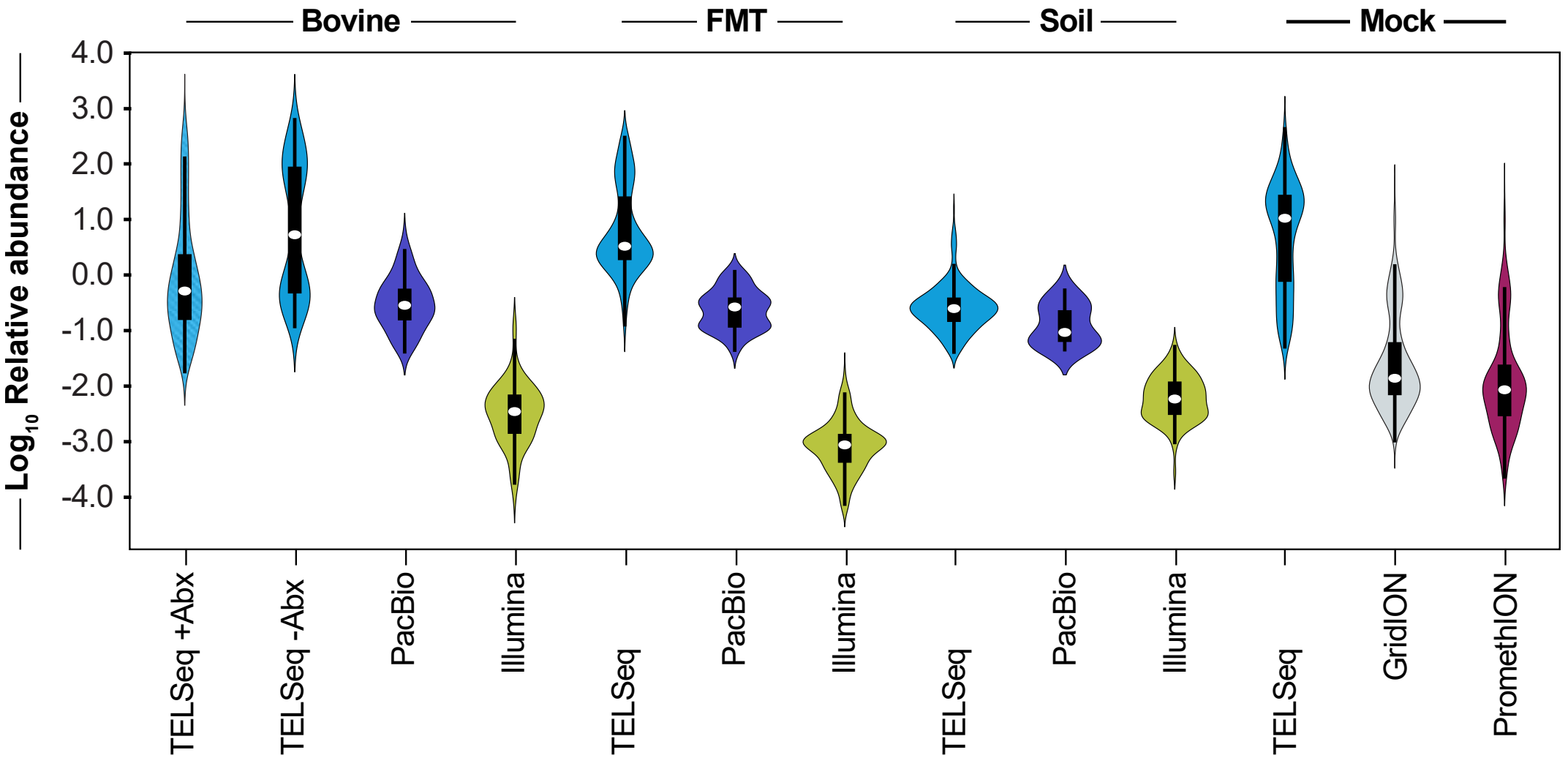


TELS I: Datasets



- Fecal Microbiota Transplant from healthy human donor **(FMT).**
- Holstein Friesian dairy cow with recent exposure to antibiotic **(+ABX).**
- Holstein Friesian dairy cow deemed systematically healthy **(-ABX).**
- Soil sample from strip of un-utilized prairie in Mower County, Minnesota **(SOIL).**
- ZymoBiomics™ microbial community **(MOCK).**

Log Relative Abundance of ARGs

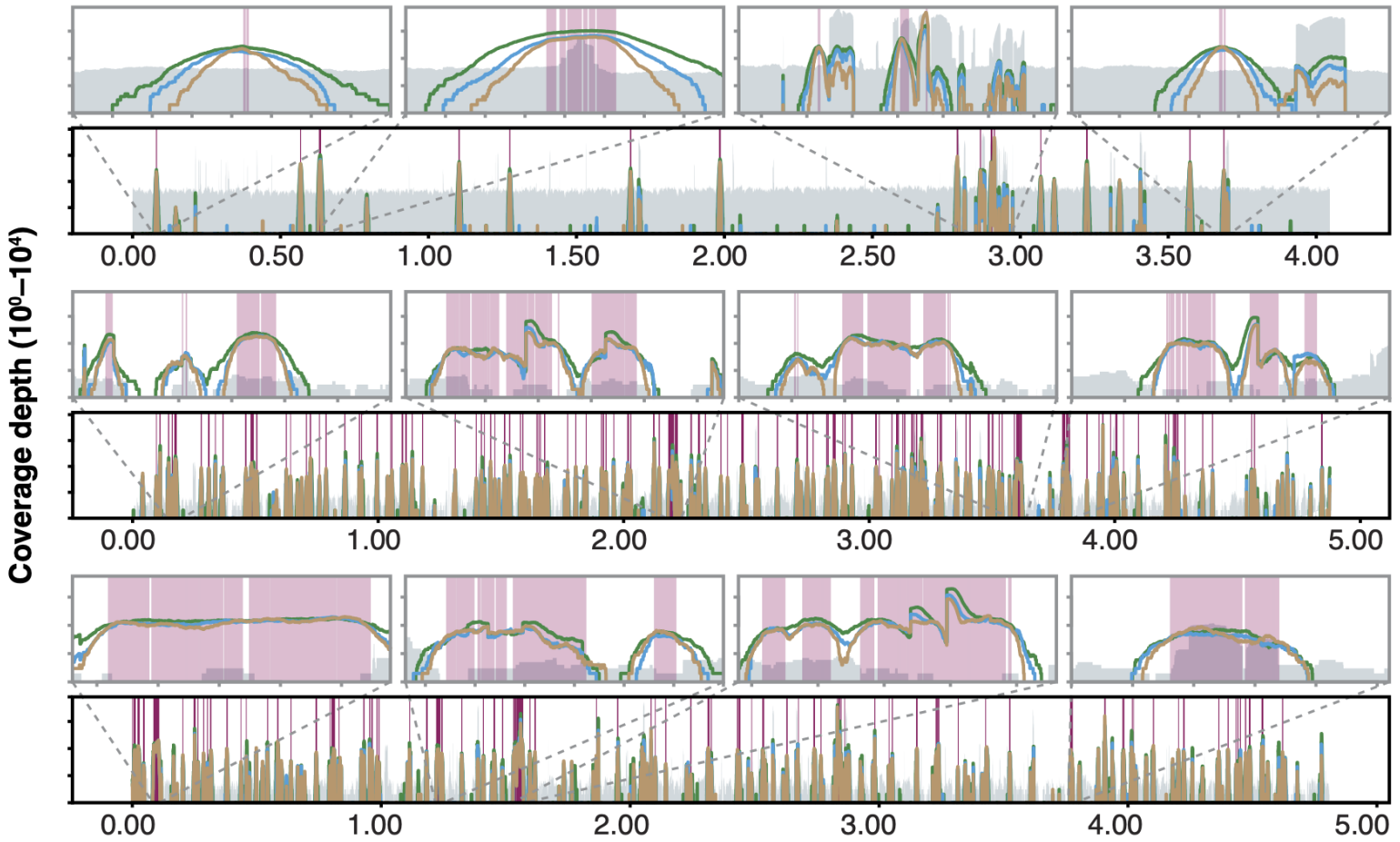


Coverage Plots

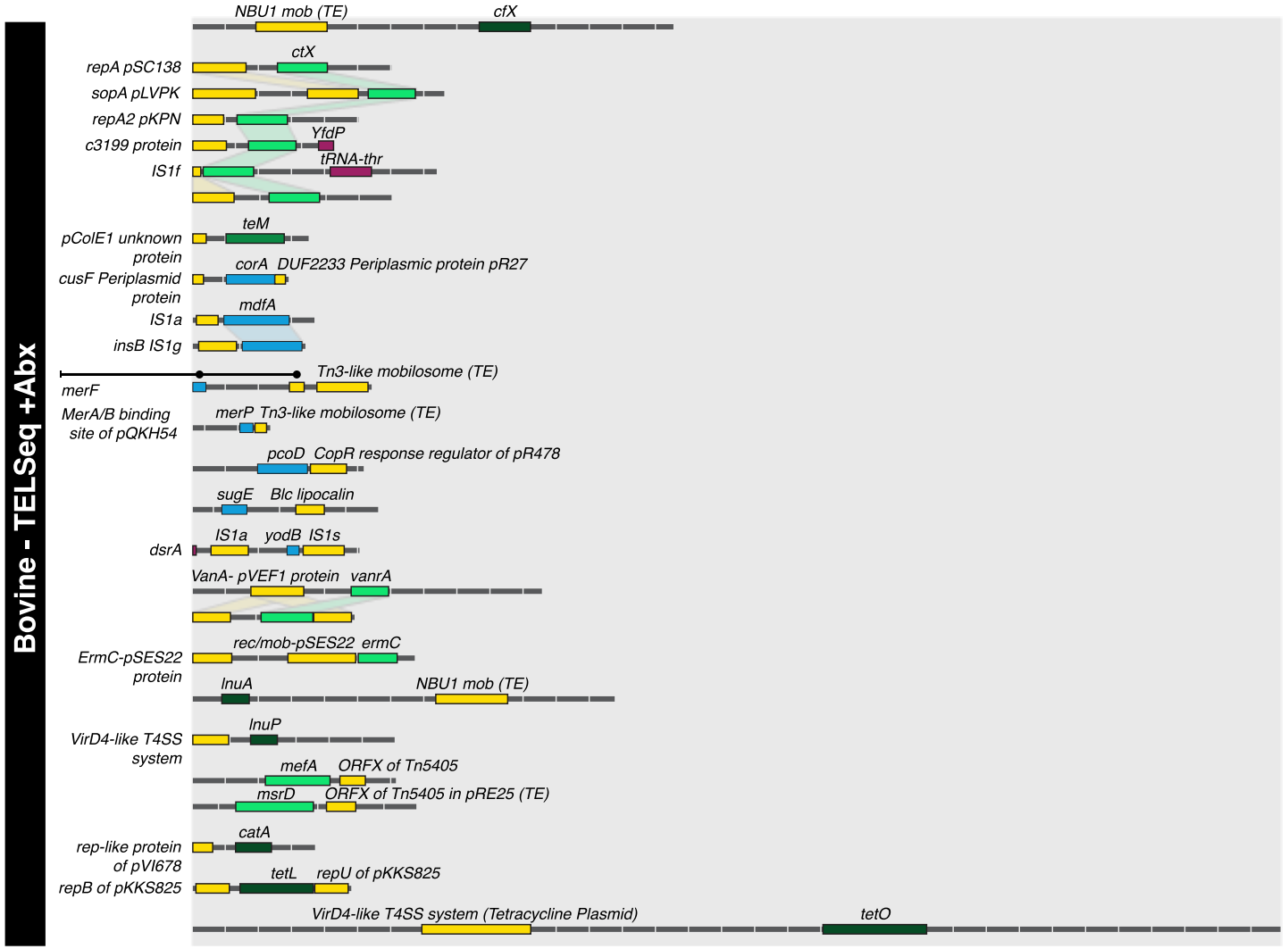
c. *Bacillus subtilis*
00.890000%

d. *Escherichia coli*
00.089000%

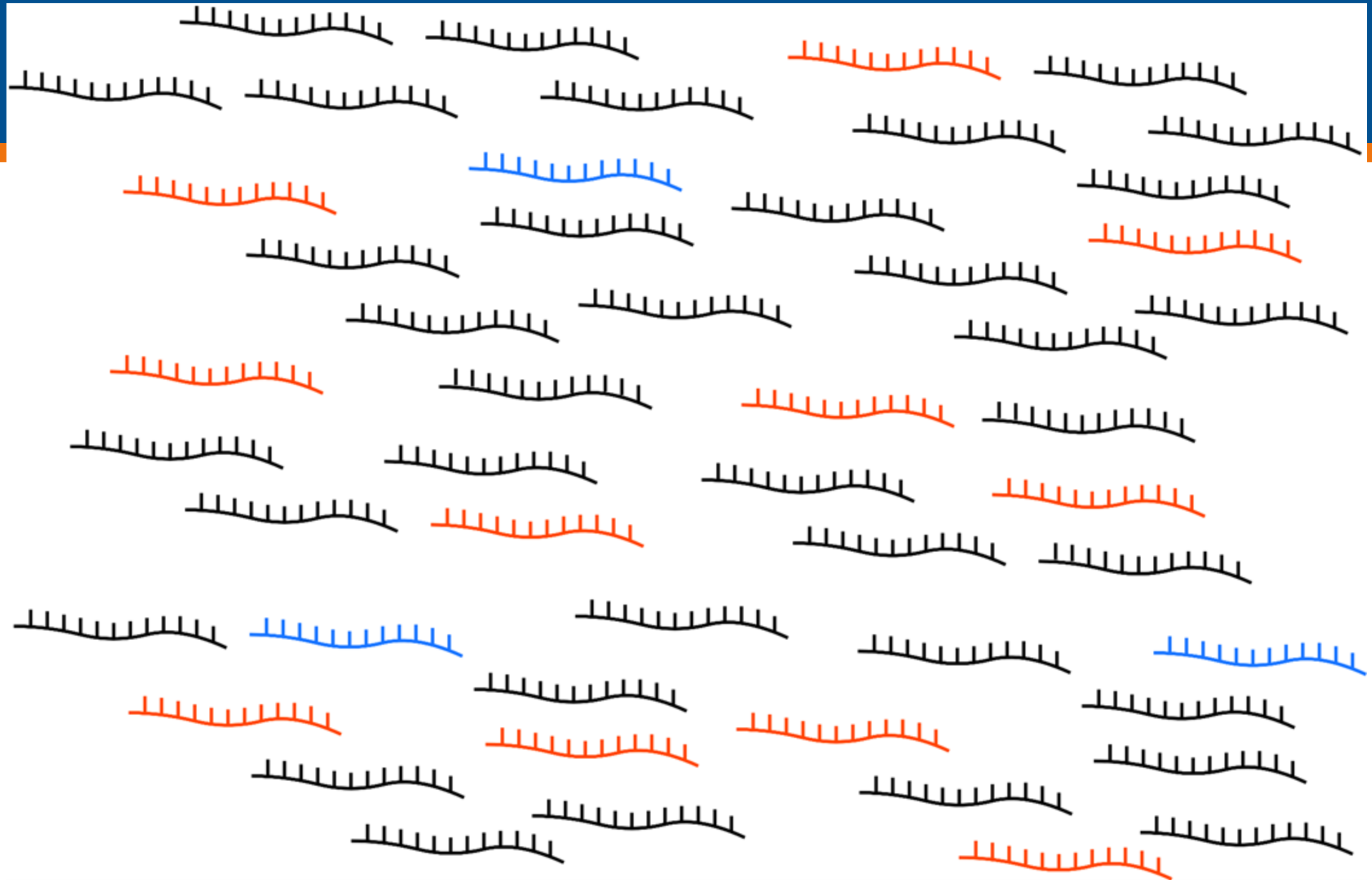
e. *Salmonella enterica*
00.089000%

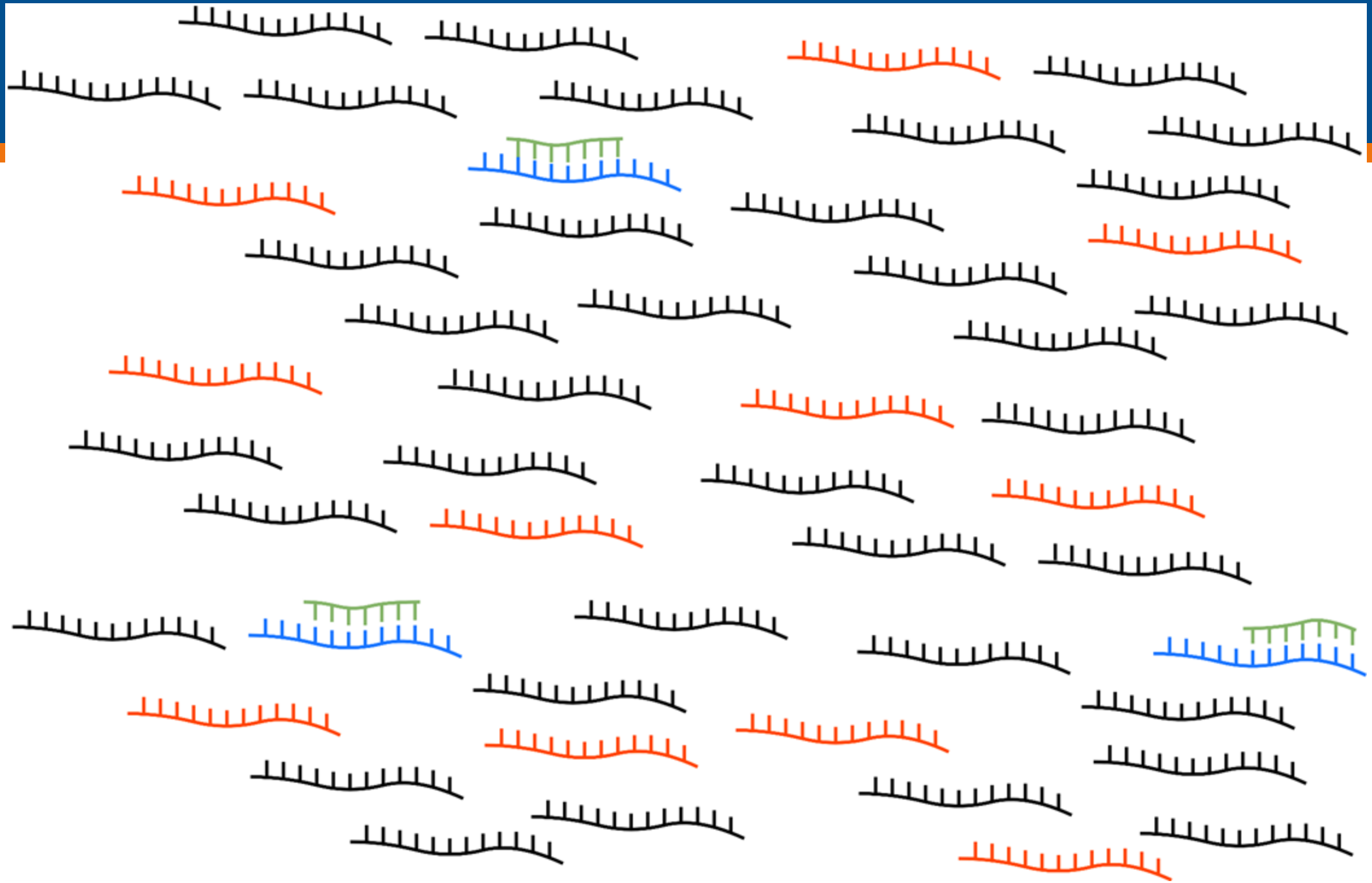


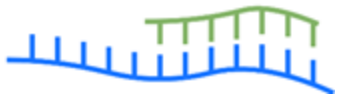
Read Annotation



Amplification of Resistance







Manufacturing Costs

Tier 1: 1- 499 kbp (up to 57,606 baits) **$12,796.00**

Tier 2: 0.5 - 2.999 Mbp (up to 57,606 baits) **$16,560.00**

Tier 3: 3.0 - 5.999 Mbp (up to 57,606 baits) **$21,578.00**

Tier 4: 6.0 - 11.999 Mbp (up to 115,212 baits) **$32,517.50**

Tier 5: 12.0 - 24.0 Mbp (up to 230,424 baits) **$40,546.50**

Manufacturing Costs

Tier 1: 1- 499 kbp (up to 57,606 baits) **$12,796.00**

Tier 2: 0.5 **16,560.00**

Tier 3: 3.0 **21,578.00**

Tier 4: 6.0) **$32,517.50**

Tier 5: 12.0 - 24.0 Mbp (up to 230,424 baits) **$40,546.50**

Other Considerations:

Coverage, Specificity, GC-content, Blacklisting contaminants,

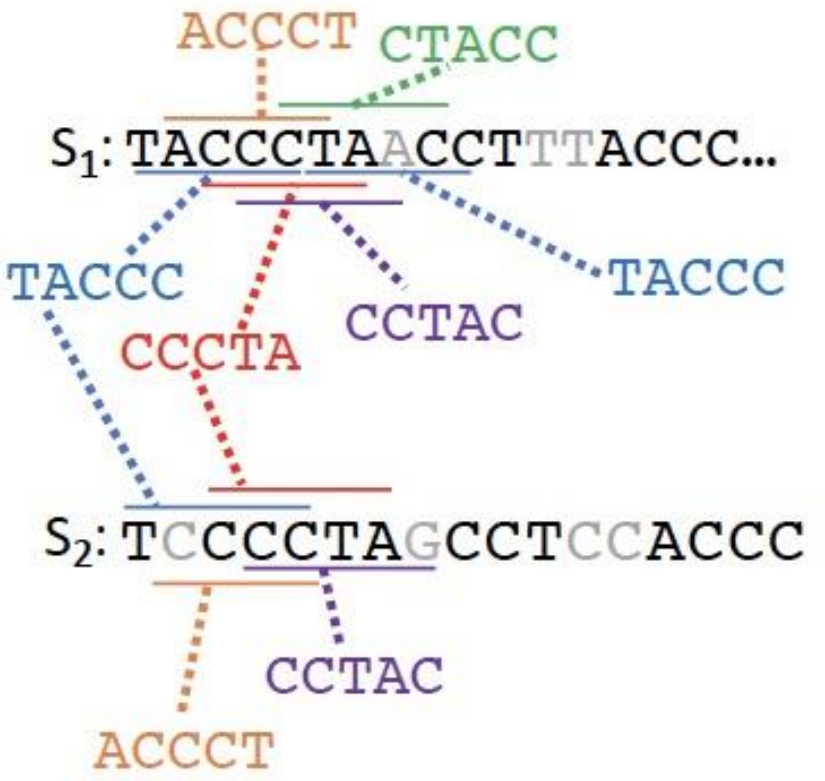
I can create synthetic probes and apply them to enrich for target DNA!!

Great! I can run a k-mer counter on the database! That's easy.

Well, the probes bind to anything with 70% sequence homology and I need the smallest set possible…

Ugh.

The Minimum Bait Cover Problem



b_1: TACCC
b_2: ACCCT
b_3: CCCTA
b_4: CCTAC
b_5: CTACC

A set $T = \{T_1, .., T_m\}$ θ-covers $S = \{S_1, \ldots, S_n\}$ if for every S_i there exists at least one in T that has Hamming distance at most θ from S_i

The Minimum Bait Cover Problem

Input: two integers $\theta \geq 0$ and $L > 0$ and a set S of n strings $S_1, \ldots, S_n$ over a finite alphabet Σ.

Question: What is the smallest possible set T of L –length strings such that S is θ-covered by T.

Minimum Bait Cover in Hard

Proposition 1: Minimum Bait Cover is NP-hard, even for $n = 1$, $\theta = 1$, and $L = 2$.

- Rules out an FPT or Slicewise Polynomial algorithm w.r.t. n, θ, L.
- However, this result requires that $|\Sigma|$ is large.

Minimum Bait Cover in Hard... Very Hard

Theorem 2. For every $k \geq 2$, Minimum Bait Cover is NP-hard even for $|\Sigma| = k$, $S = \{S\}$, $\theta = 0$ and $L = O(\log |S|)$.

- The problem remains intractable for small problem instances, e.g., DNA alphabet, a single sequence, Hamming distance of 0, and $\log |S|$
- Solves an existing open problem in the literature.

Heuristic Solution

- Build the de Bruijn graph, find all the unitigs, find all L length sequences on unitigs.
- Slide a window of length L. If the starting position is not yet covered, take it; if not go the next one.
- Dynamic programming algorithms to bootstrap a solution
- Expectation-Maximization algorithms.

Heuristic Solution

1. Construct the FM-index and GSA of $S_1,\ldots,S$
2. Store a bit vector for each position of $S_1,\ldots,S_n$
3. Consider each position of every string in $S_1,\ldots,S_n$, say $S_i[j]$ if it not covered then we add the L-mer starting at $S_i[j]$ to the bait set.
4. Use the FM-index and GSA to find all positions covered by the newly added bait.

Heuristic Solution

1. Construct the FM-index and GSA of $S_1,\ldots,S_n$
2. Store a bit vector for each position of $S_1,\ldots,S_n$
3. Consider each position of every string in $S_1,\ldots,S_n$, say $S_i[j]$ if it not covered then we add the L-mer starting at $S_i[j]$ to the bait set.
4. Use the FM-index and GSA to find all positions covered by the newly added bait.

Heuristic Solution

```
For i = 1..|B| - k + 1 do:
    x ← B[i..i+k-1]
    [r1, r2] ← Fmindex.Search(x)

    For j = r1..r2 do
        t, p ← GSA[j]
        B′ ← St[p-i+1..p-i+L]

        if d(B′, B) ≤ θ
            Mark St[p-i+1..p-i+L] as covered
```

Find all k-mers

Extend to k-mers to length L

Competing Methods

- **CATCH** (Nature Biotech. 2019) has quadratic running time in practice.
- **MrBait** requires all-pairs alignment of baits as post-processing.
- **BaitFisher** requires a multiple sequence alignment.

GATACCGATTACCGATGGAAGA

TGGGGAAAGATACCGATTTCCG

ATGGAAGATGGAATTACTCAGA

GACAGGGAAATAAGTCCTG

GATACCGATTACCGATGGAAGA

TGGGGAAAGATACCGATTTCCG

ATGGAAGATGGAATTACTCAGA

GACAGGGAAATAAGTCCTG

GATACCGATTACCGATGGAAGA

TGGGGAAAGATACCGATTTCCG

ATGGAAGATGGAATTACTCAGA

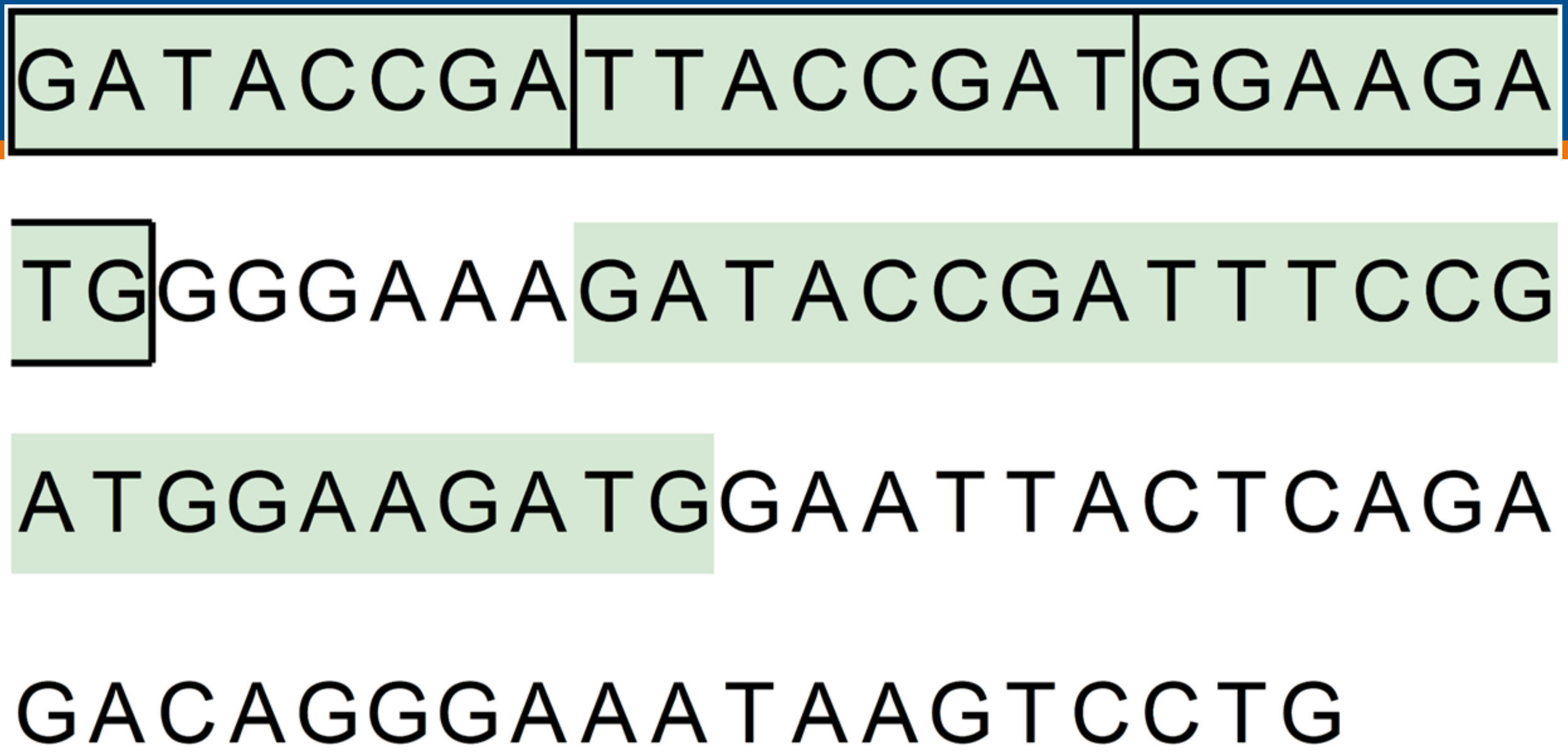
GACAGGGAAATAAGTCCTG

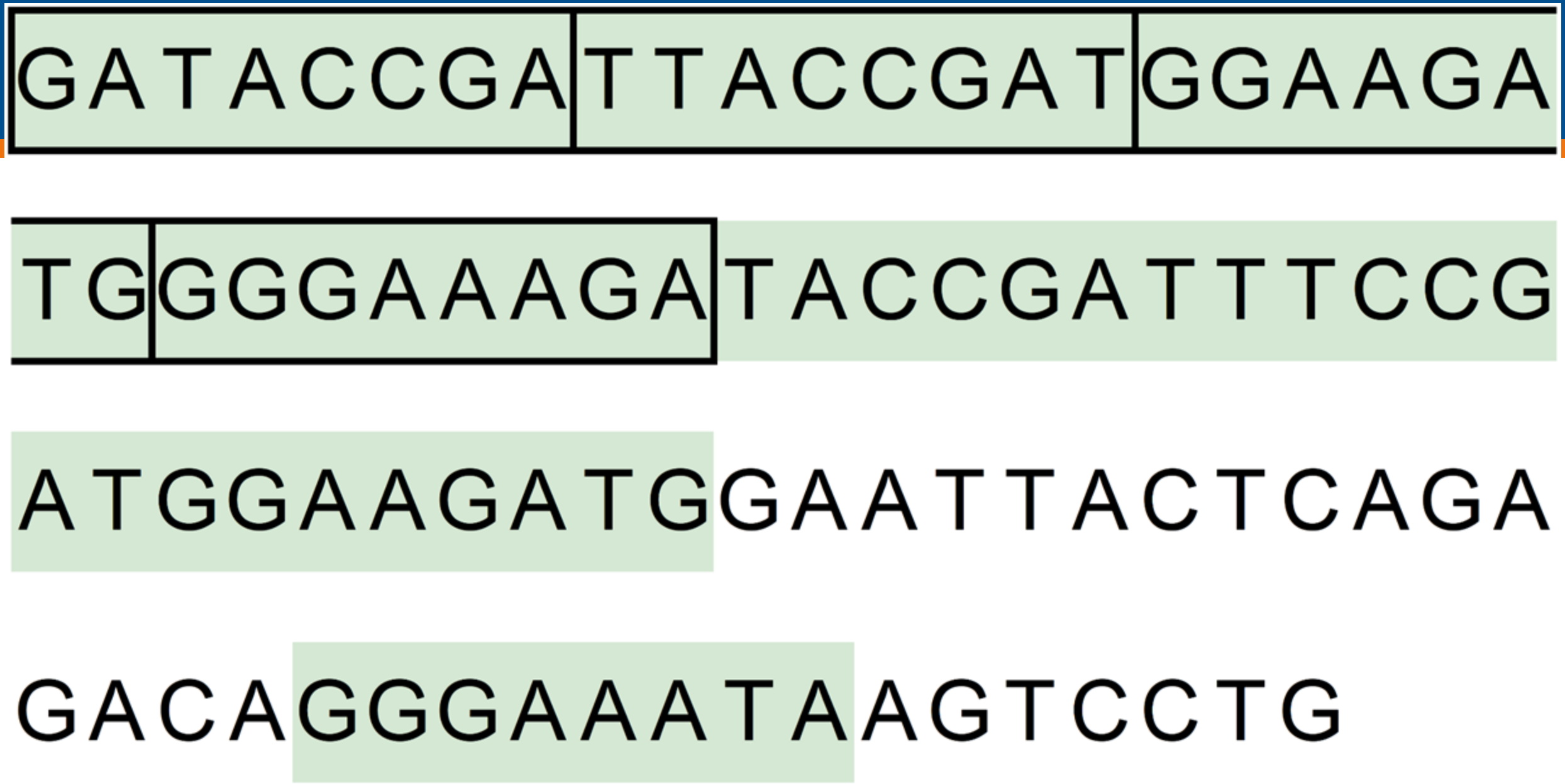
GATACCGATTACCGATGGAAGA

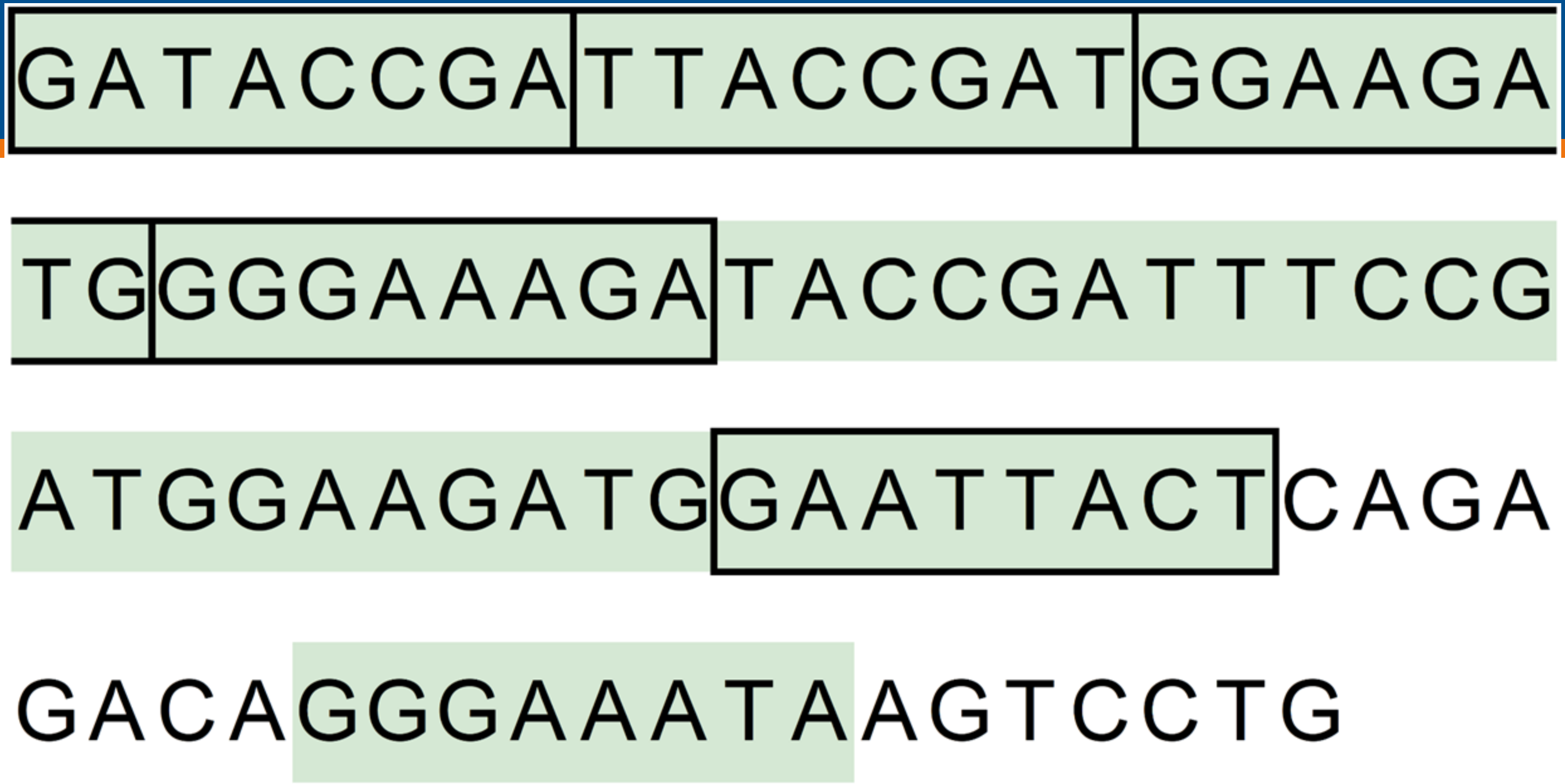
TGGGGAAAGATACCGATTTCCG

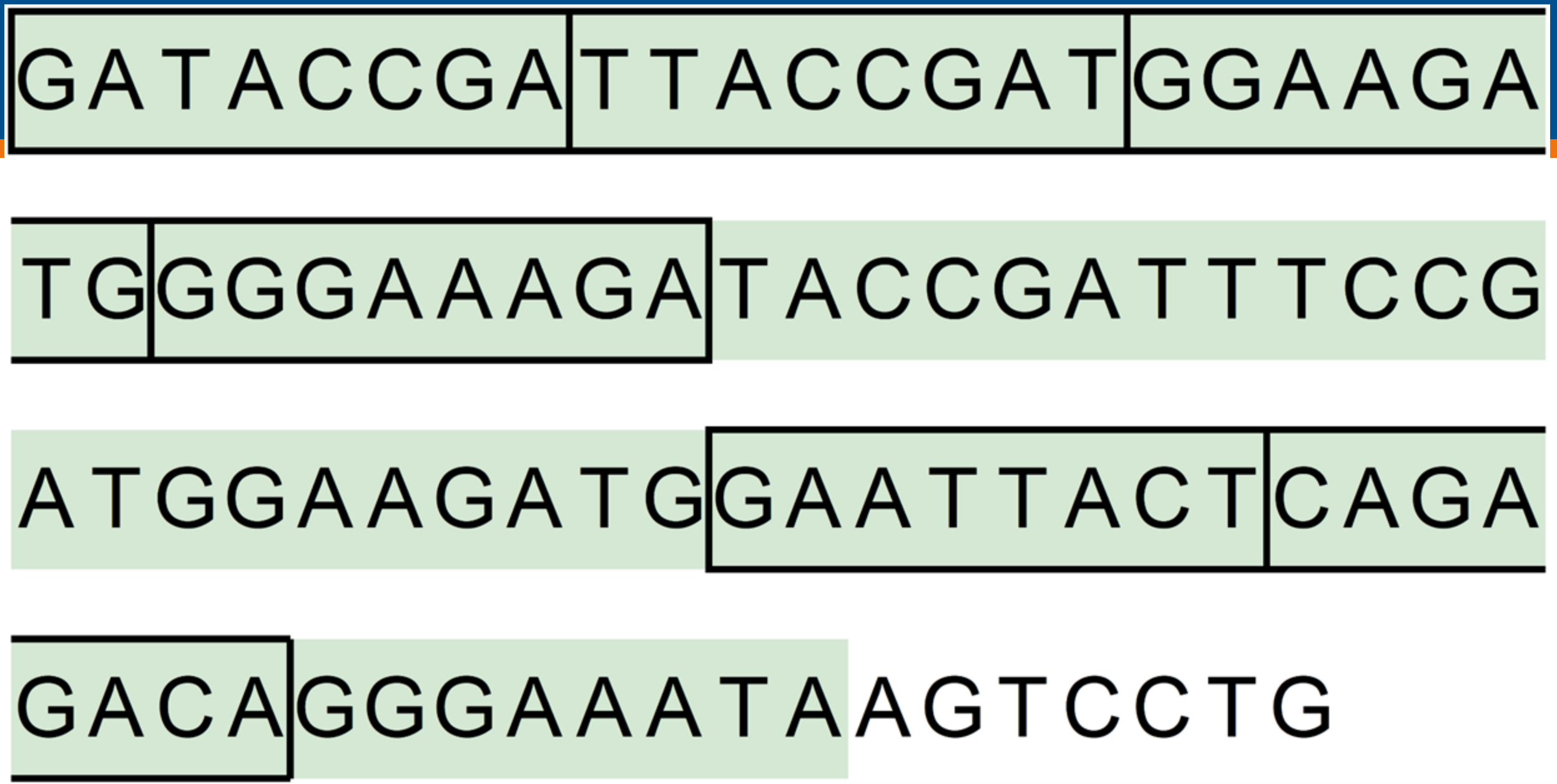
ATGGAAGATGGAATTACTCAGA

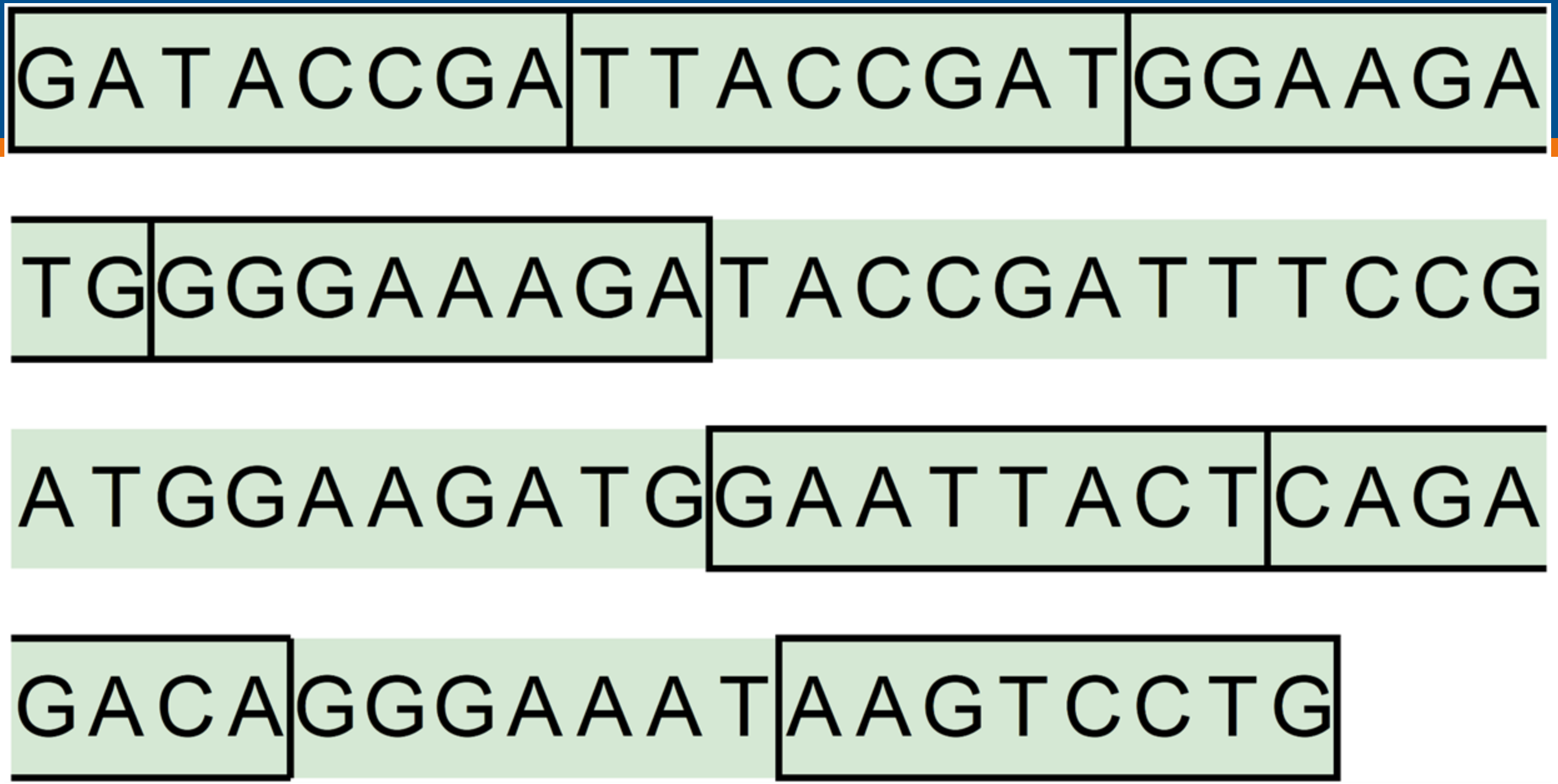
GACAGGGAAATAAGTCCTG


GATACCGATTACCGATGGAAGA
TGGGGGAAAGATACCGATTTCCG
ATGGAAGATGGAATTACTCAGA
GACAGGGAAATAAGTCCTG


GATACCGA TTACCGAT GGAAGA
TG GGGGAAAGA TACCGATTTCCG
ATGGAAGATG GAATTACTCAGA
GACA GGGAAATA AGTCCTG


GATACCGA TTACCGAT GGAAGA
TG GGGGAAAGA TACCGATTTCCG
ATGGAAGATG GAATTACT CAGA
GACA GGGAAATA AGTCCTG


GATACCGA TTACCGAT GGAAGA
TG GGGGAAAGA TACCGATTTCCG
ATGGAAGATG GAATTACT CAGA
GACA GGGAAATA AGTCCTG

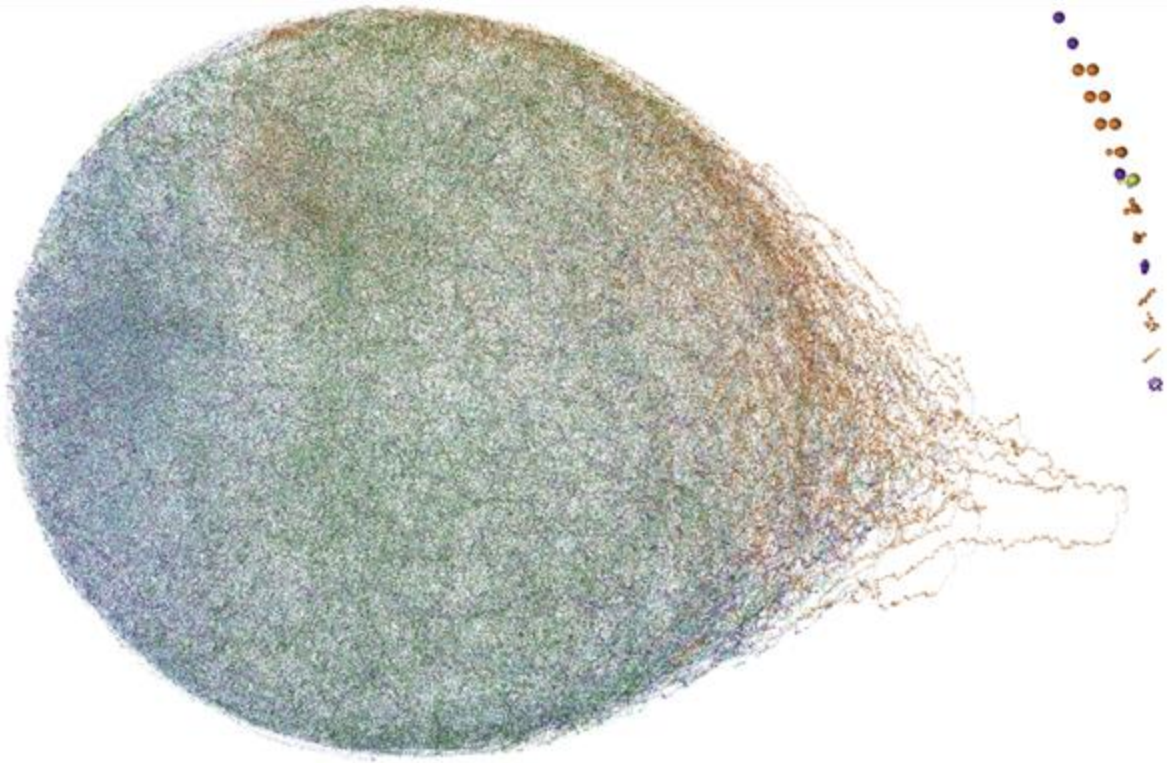

GATACCGA TTACCGAT GGAAGA
TG GGGGAAAGA TACCGATTTCCG
ATGGAAGATG GAATTACT CAGA
GACA GGGAAAT AAGTCCTG

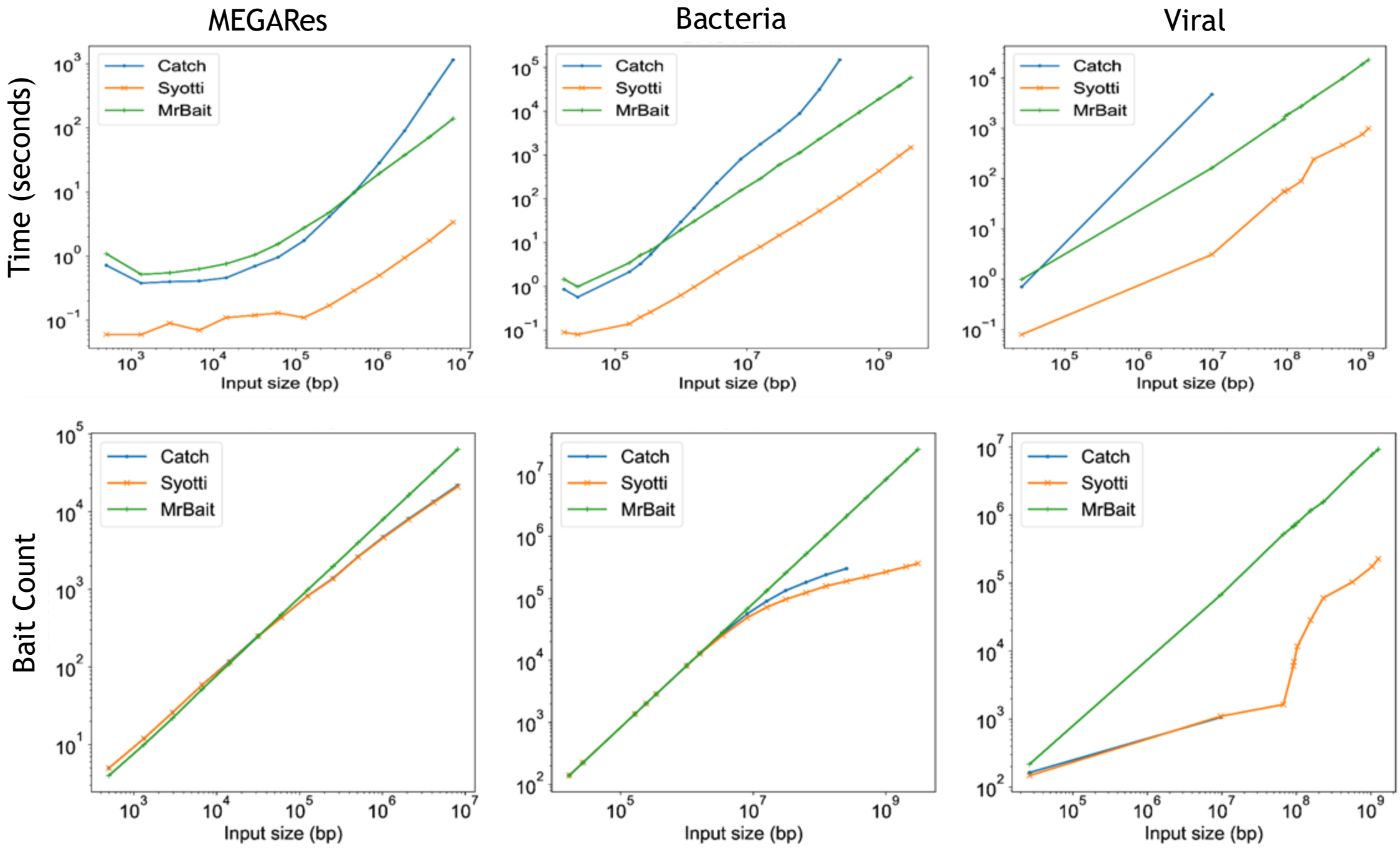
Datasets

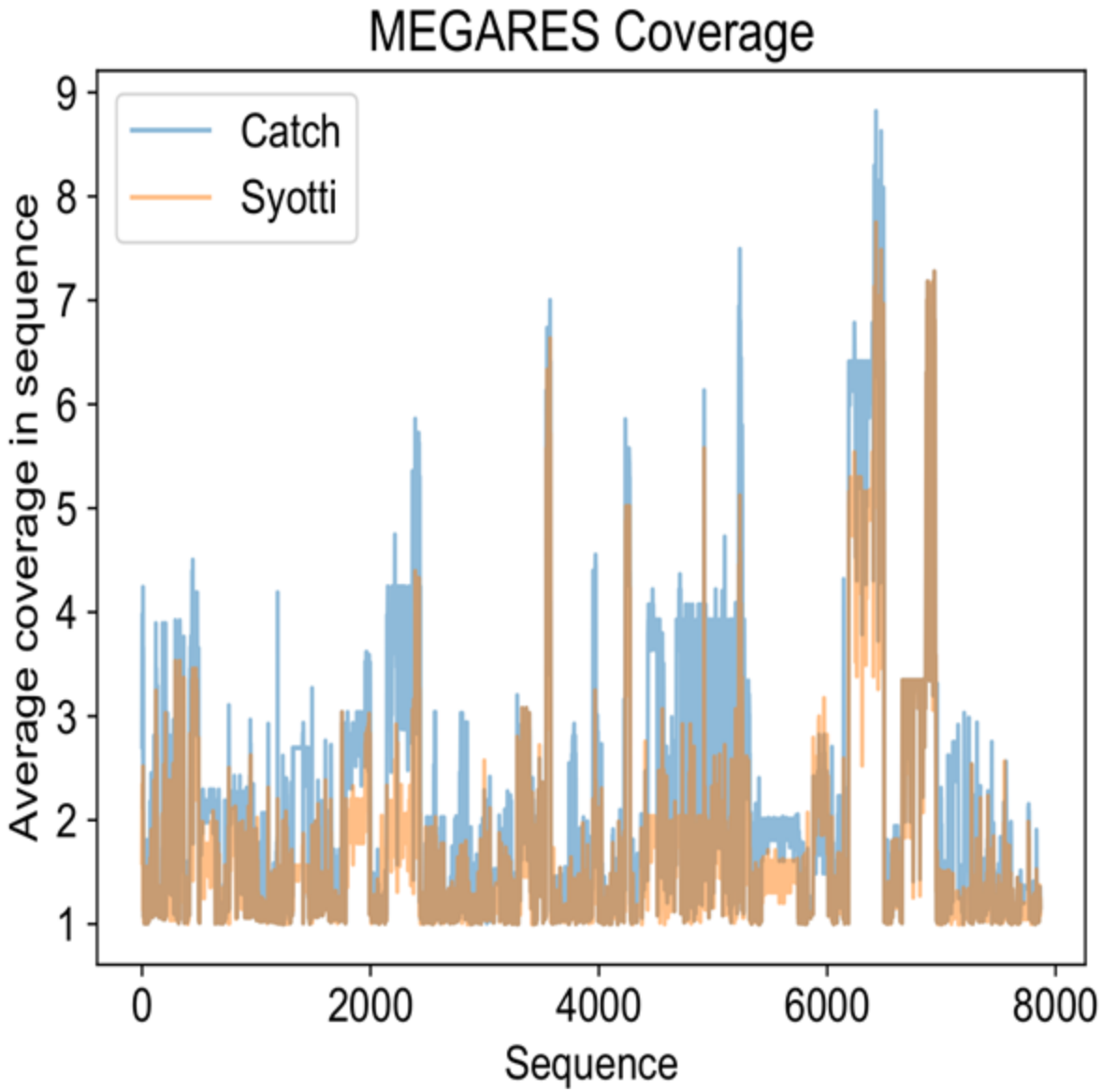
MEGARes: 7868 ARGs (**8.1 MB total)**

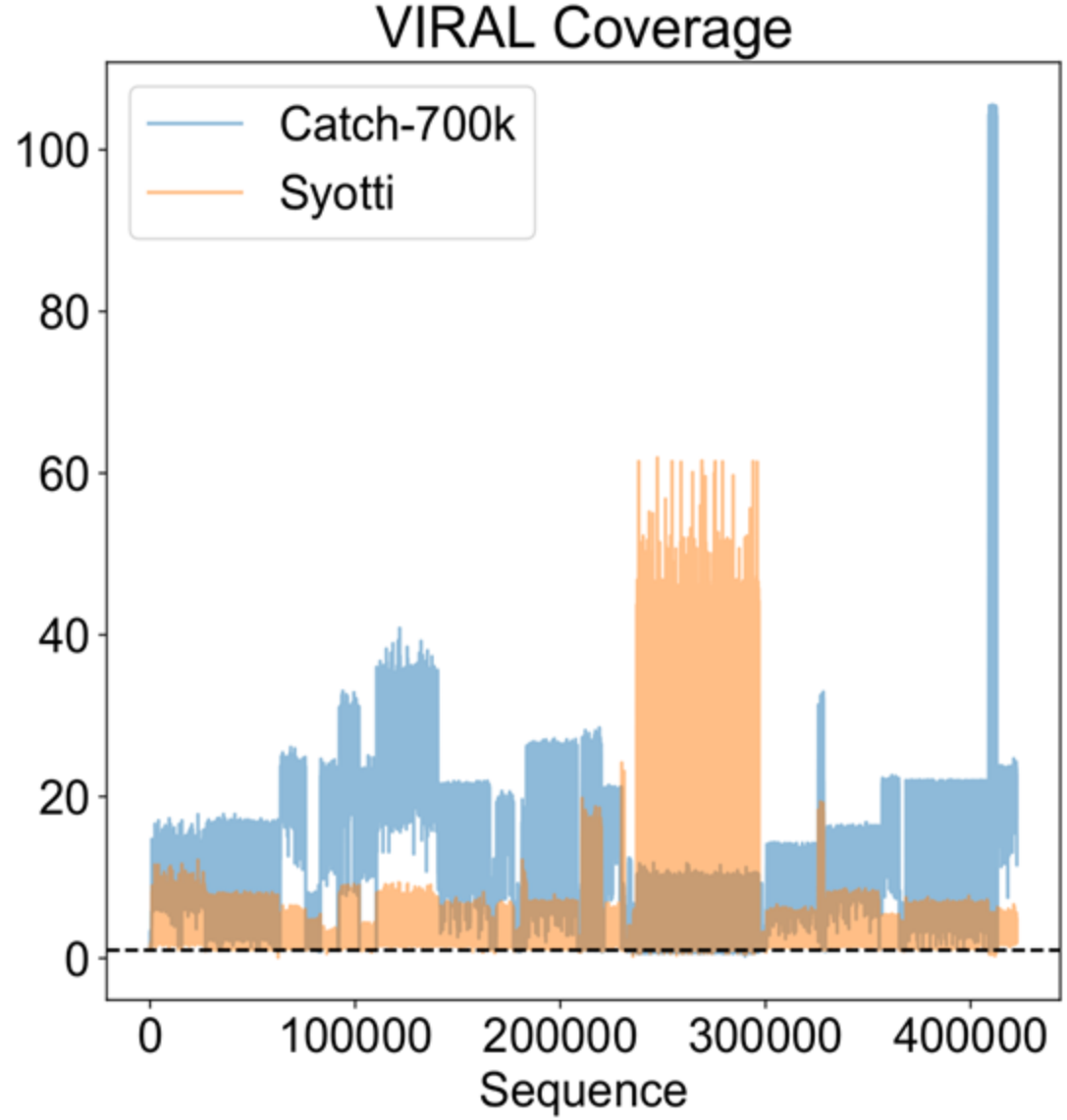
Viral: 420,000 viral genomes from 608 species (**1.3 GB total)**

Bacteria: 1,000 bacterial genomes from 4 foodborne pathogen species (**3.0 GB total)**



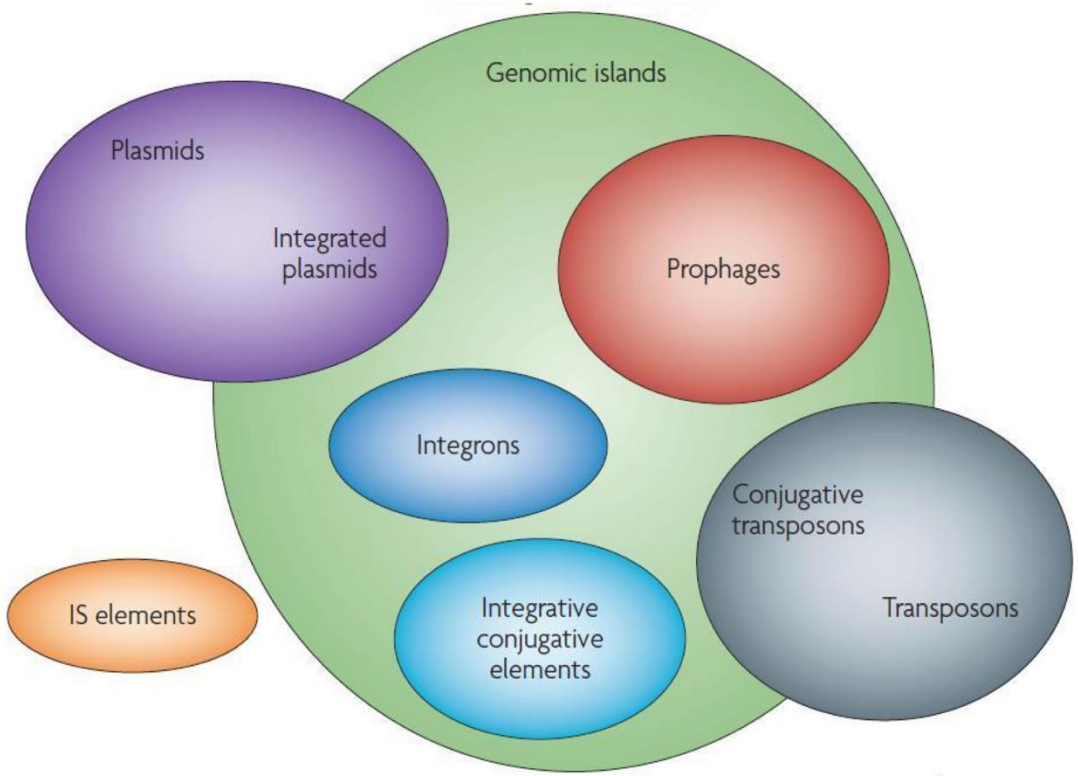

MEGARes
Bacteria
Viral
Time (seconds)
Bait Count
Catch
Syotti
MrBait
Input size (bp)


MEGARES Coverage
Catch
Syotti
Average coverage in sequence
1
2
3
4
5
6
7
8
9
0
2000
4000
6000
8000
Sequence


VIRAL Coverage
Catch-700k
Syotti
0
20
40
60
80
100
0
100000
200000
300000
400000
Sequence

Open Areas for Discovery

Detection of MGEs is Hard



Databases are overlapping and unorganized (i.e., no hierarchy).

Alignment does not yield a correct answer.

Only knowing broad information is not useful.

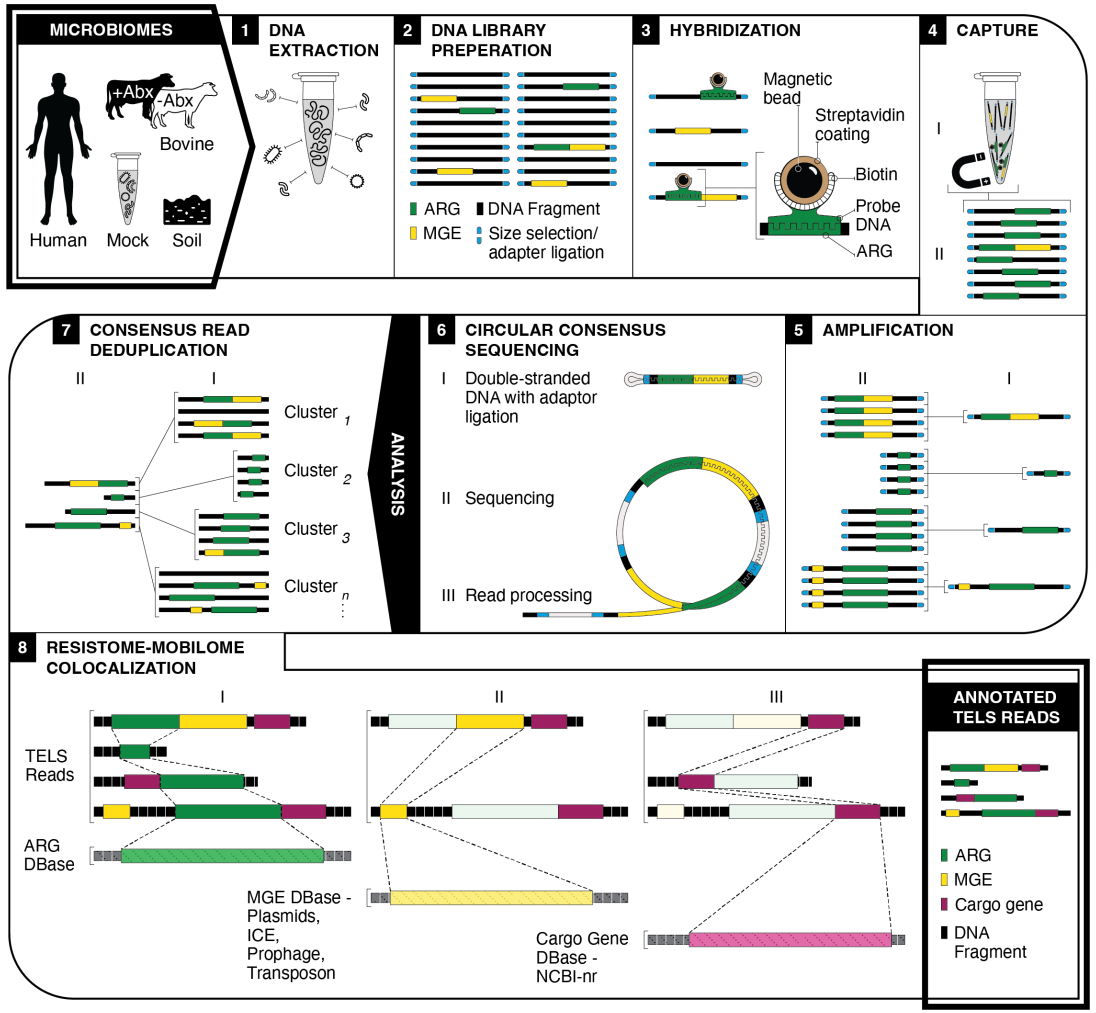
https://www.cuhk.edu.hk/med/mic/Education/SeminarFiles/2013/2013%20Dec-Wang%20Zheng.pdf

Bait Design Remains Challenging

Can the problem be reformulated?

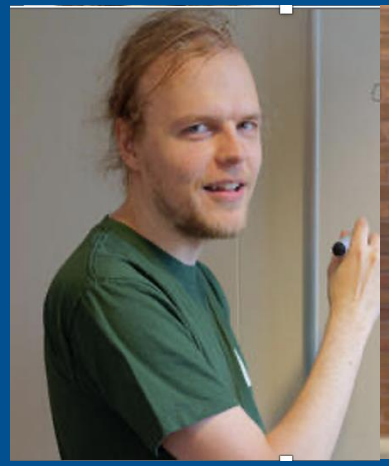
Does there exist approximate solutions?

Can off-target binding be controlled?



Thank you

Jarno Alanko



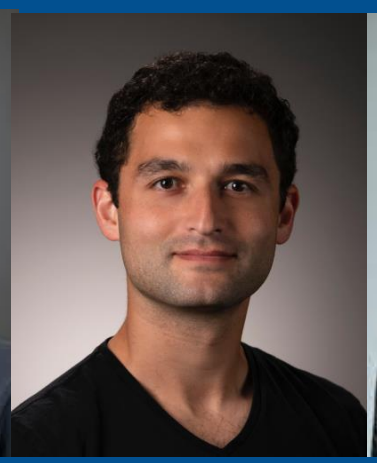
Ilya Slizovskiy

Marco Oliva

Mattia Prosperi

Travis Gagie

Dan Lokshantov

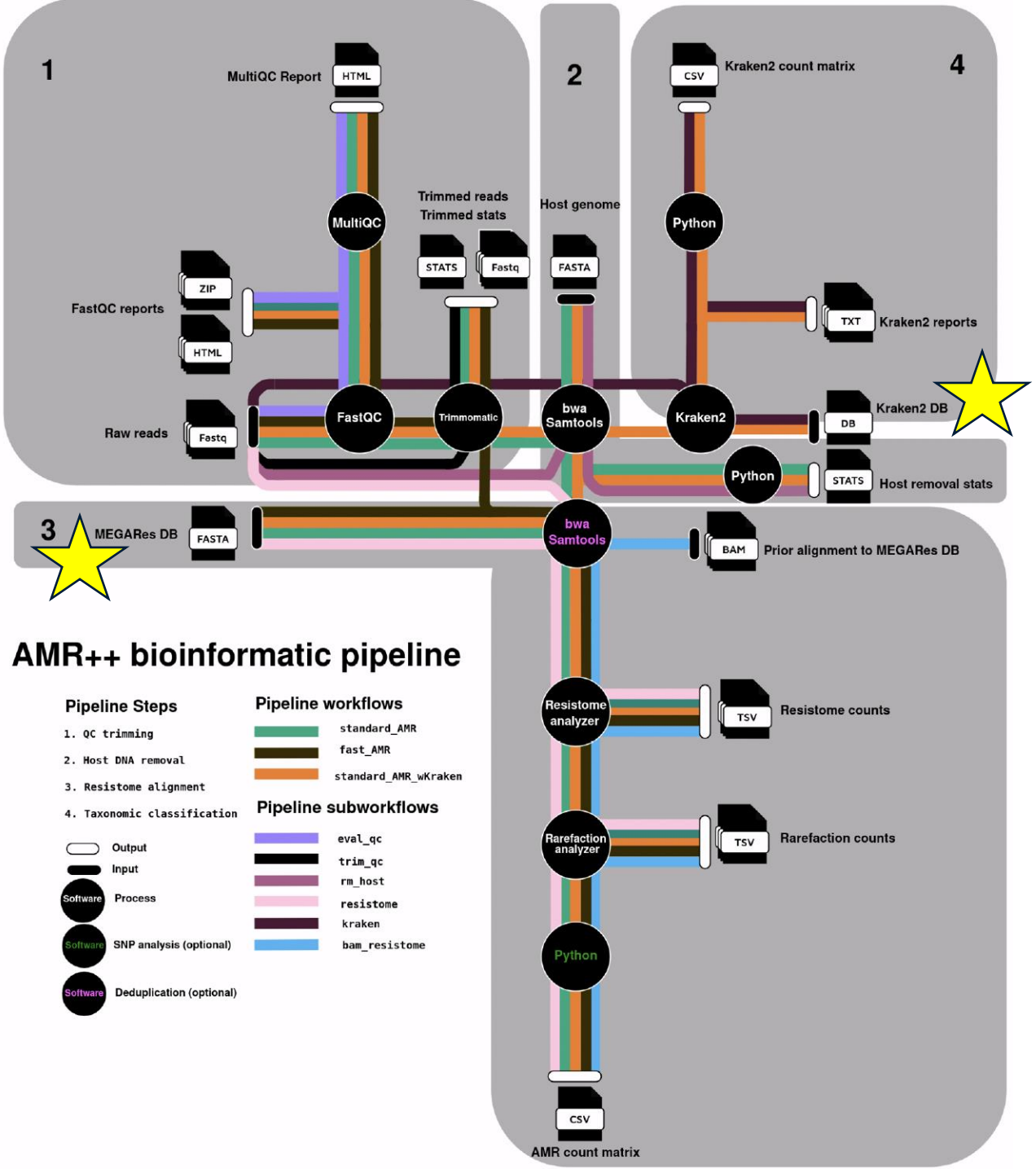


Noelle Noyes

NSF EAGER, NSF SCH, and NSF IIBR

NIH BDMA (grants 308030, 314170, and 323233)

AMR++ Pipeline

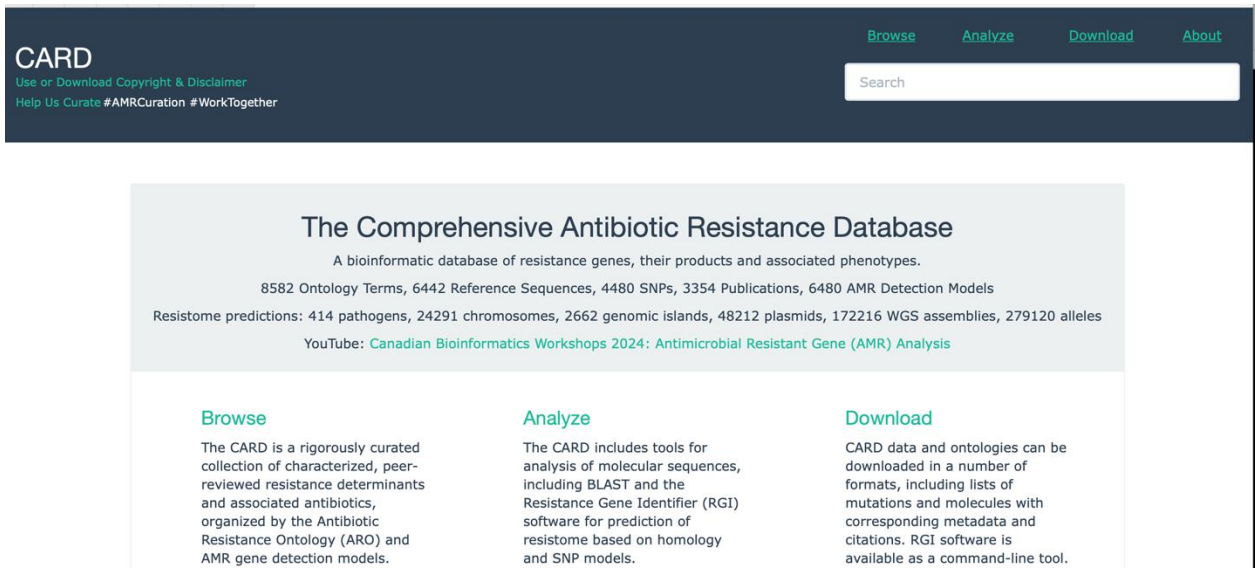

1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome

Part 1: Databases for Resistome Analysis

It’s important to understand your database

- ✓ How was the database developed?
- ✓ How “good” are its sources?
- ✓ How closely has it been checked for inaccuracies or inconsistencies?
- ✓ How comprehensive is it?

Two main databases in AMR

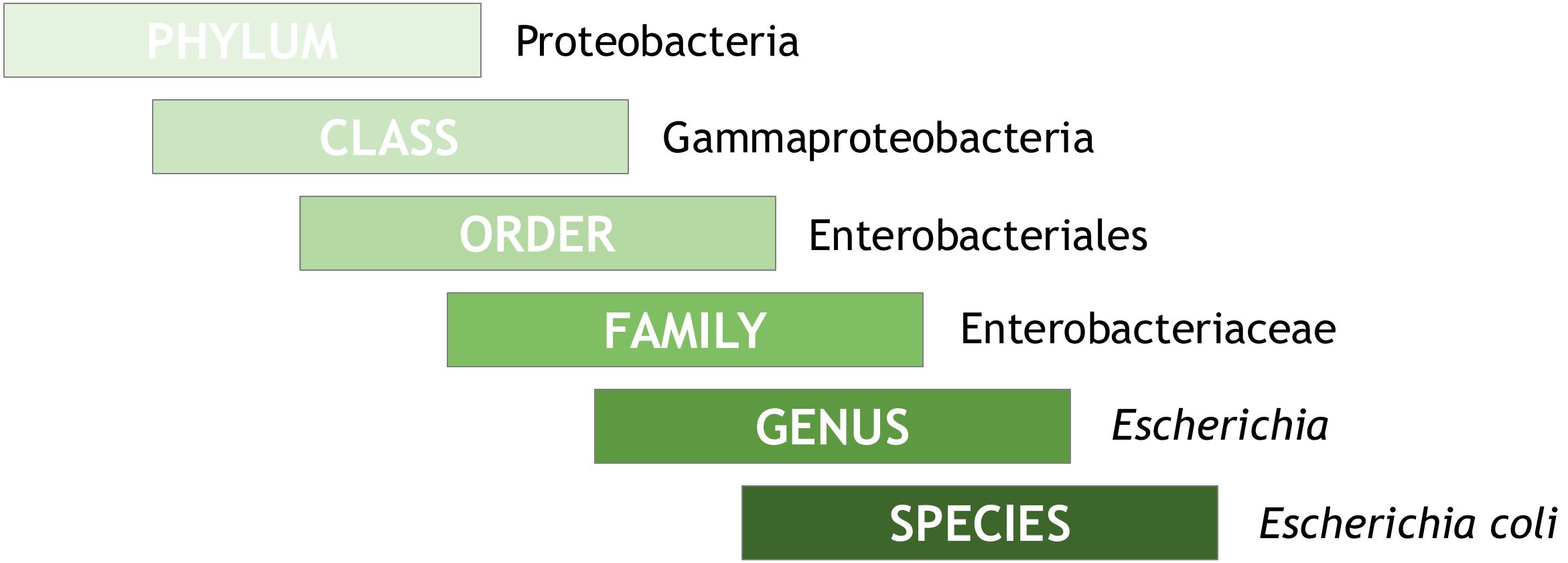


So how is MEGARes different then?

The real difference lies in the annotation structure!

(Plus, there's a lot of manual curation)

Ontology of Microbiome Data



MEGARes 3.0

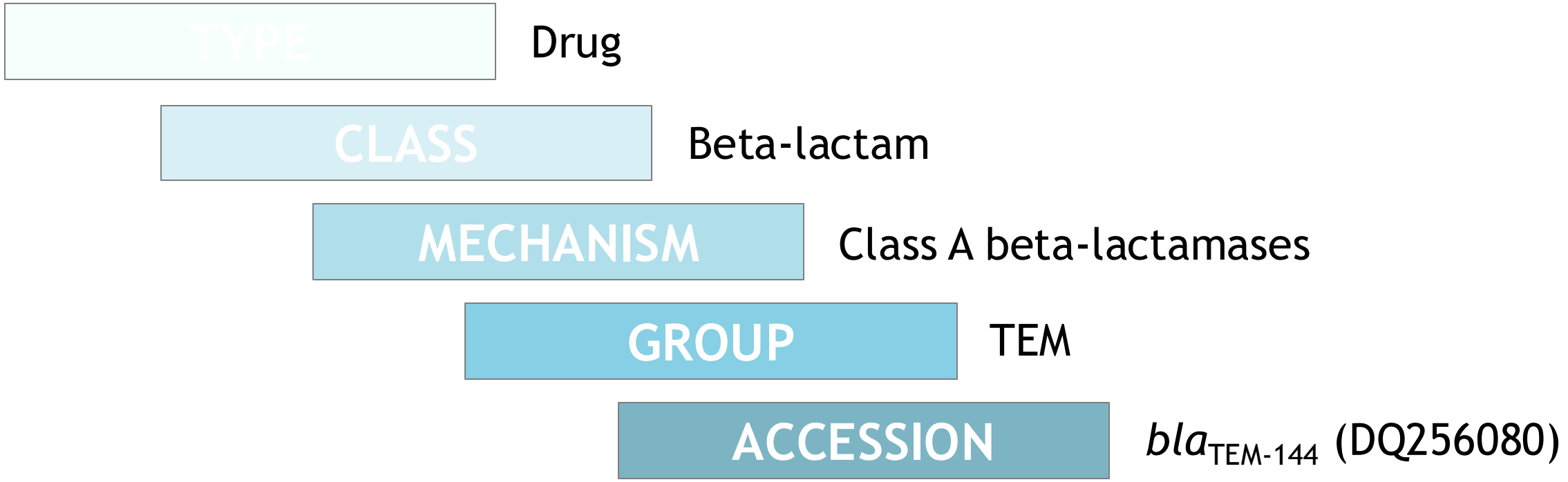
Type: the type of antimicrobial compound, e.g. drugs, biocides, multi-compound

Class: the major antimicrobial chemical class, e.g. betalactams, aminoglycosides

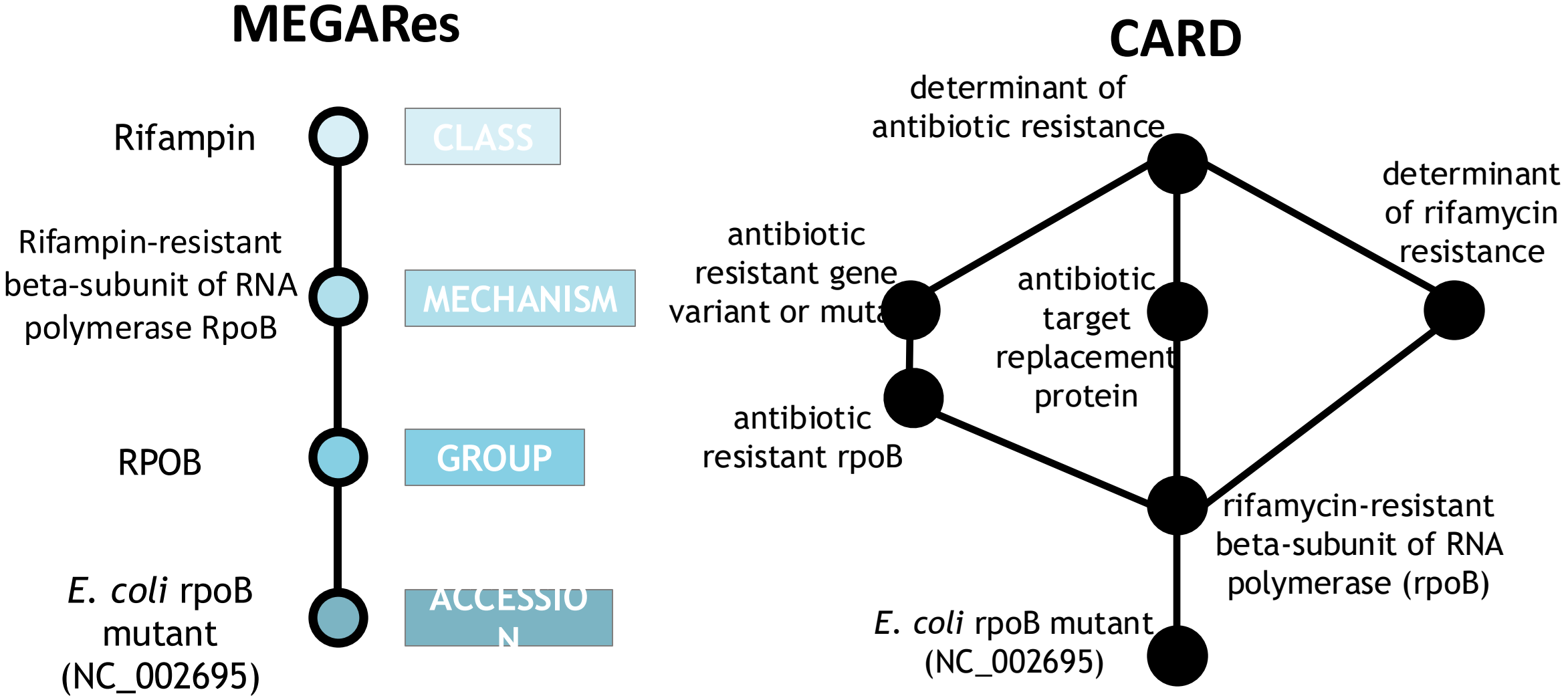
Mechanism: the biological mechanism of resistance, e.g. penicillin binding protein

Group: the gene- or operon-level group for that sequence, e.g. SHV betalactamase, MCR-1

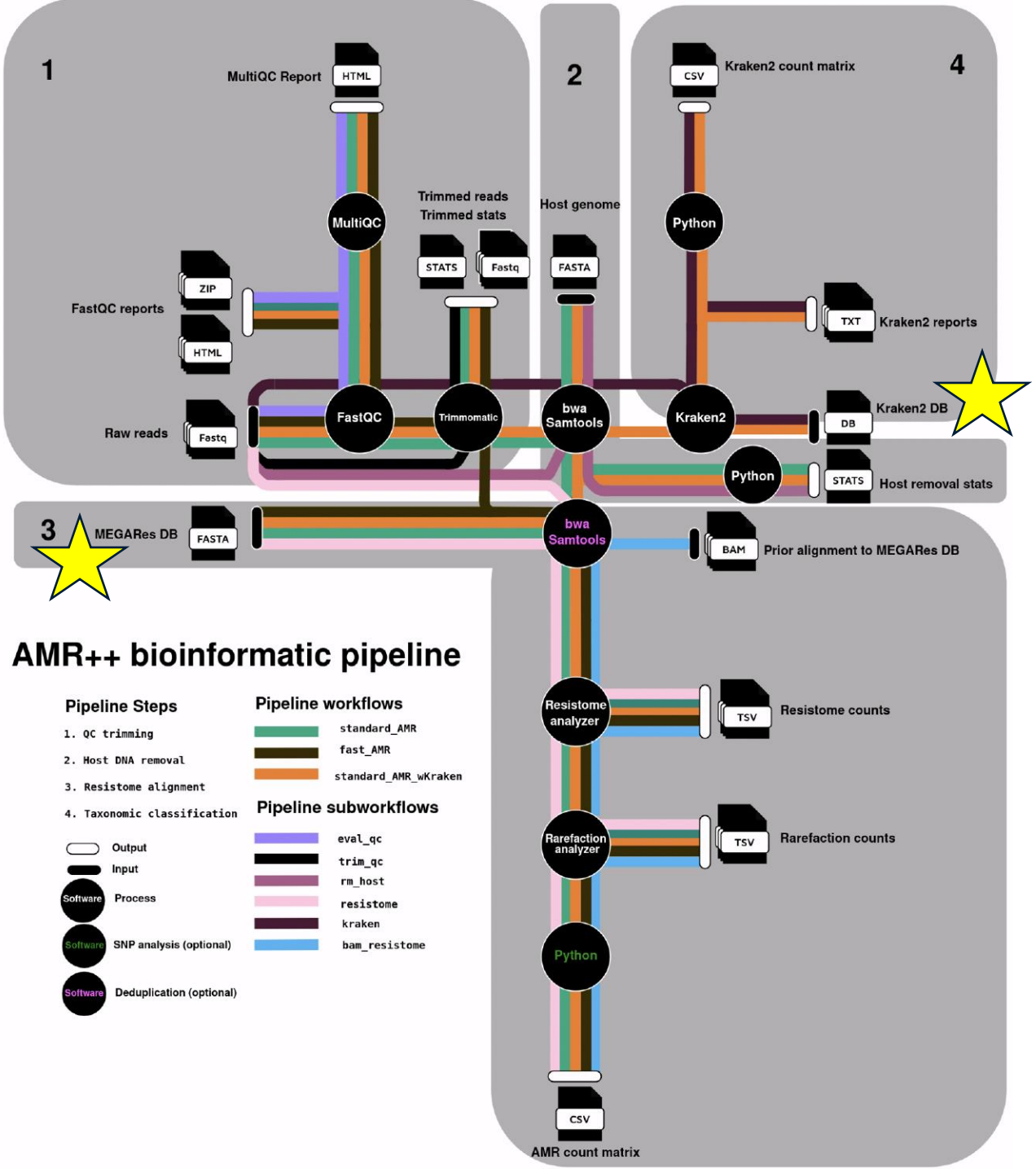
Ontology of Resistome Data

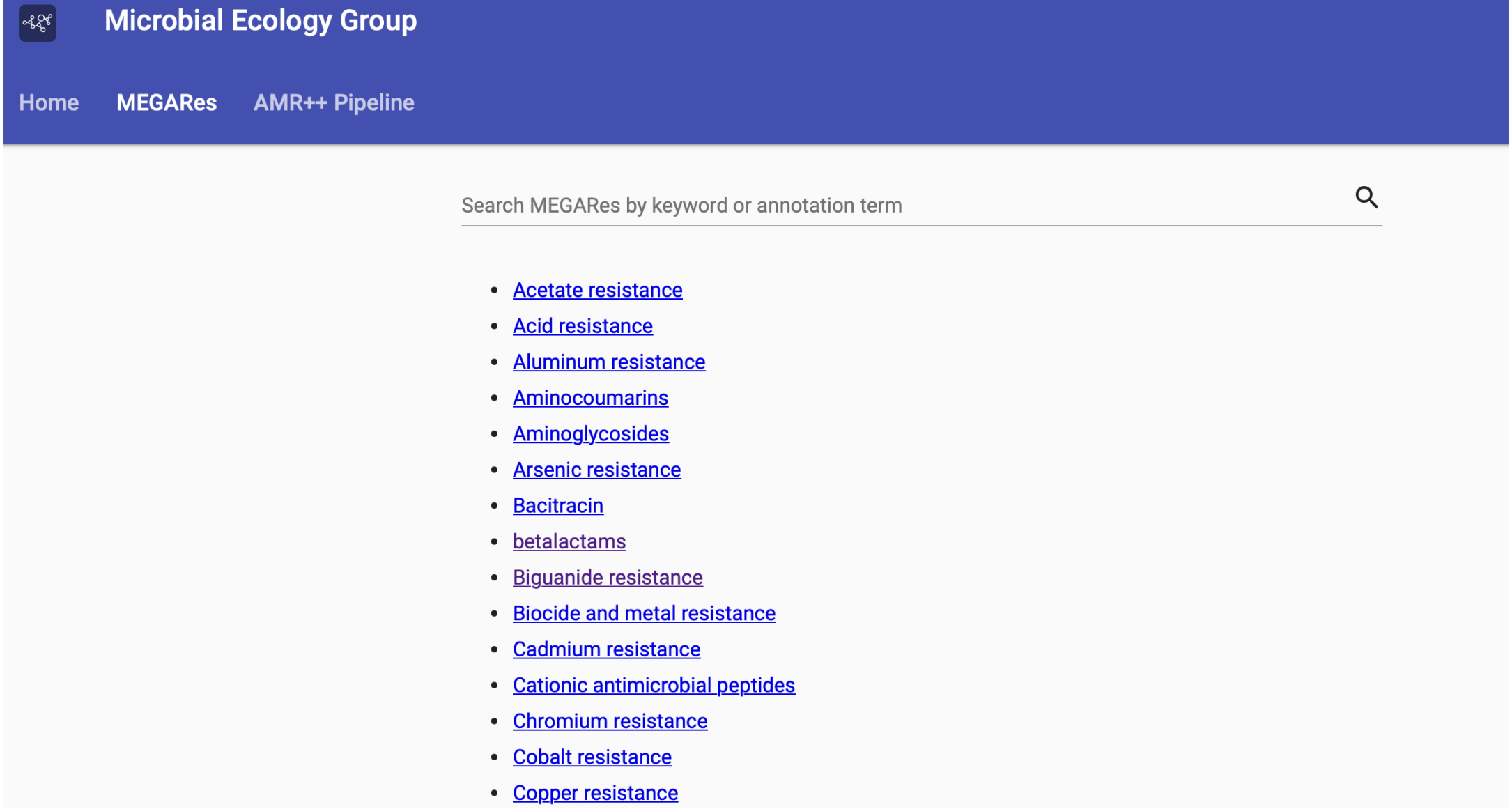


MEGARes ontology is acyclical



Comprehensive Antimicrobial Resistance Database, McArthur et al., 2013


1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome


Microbial Ecology Group
Home
MEGARes
AMR++ Pipeline
Search MEGARes by keyword or annotation term
Acetate resistance
Acid resistance
Aluminum resistance
Aminocoumarins
Aminoglycosides
Arsenic resistance
Bacitracin
betalactams
Biguanide resistance
Biocide and metal resistance
Cadmium resistance
Cationic antimicrobial peptides
Chromium resistance
Cobalt resistance
Copper resistance

Multi-drug resistance

We define multi-drug resistance as genes and mechanisms that cause resistance to two or more different antibiotic classes. Typically, such mechanisms involve active extrusion of antibiotic molecules from the bacterial cell or mechanisms that prevent the drug from reaching its target.

References

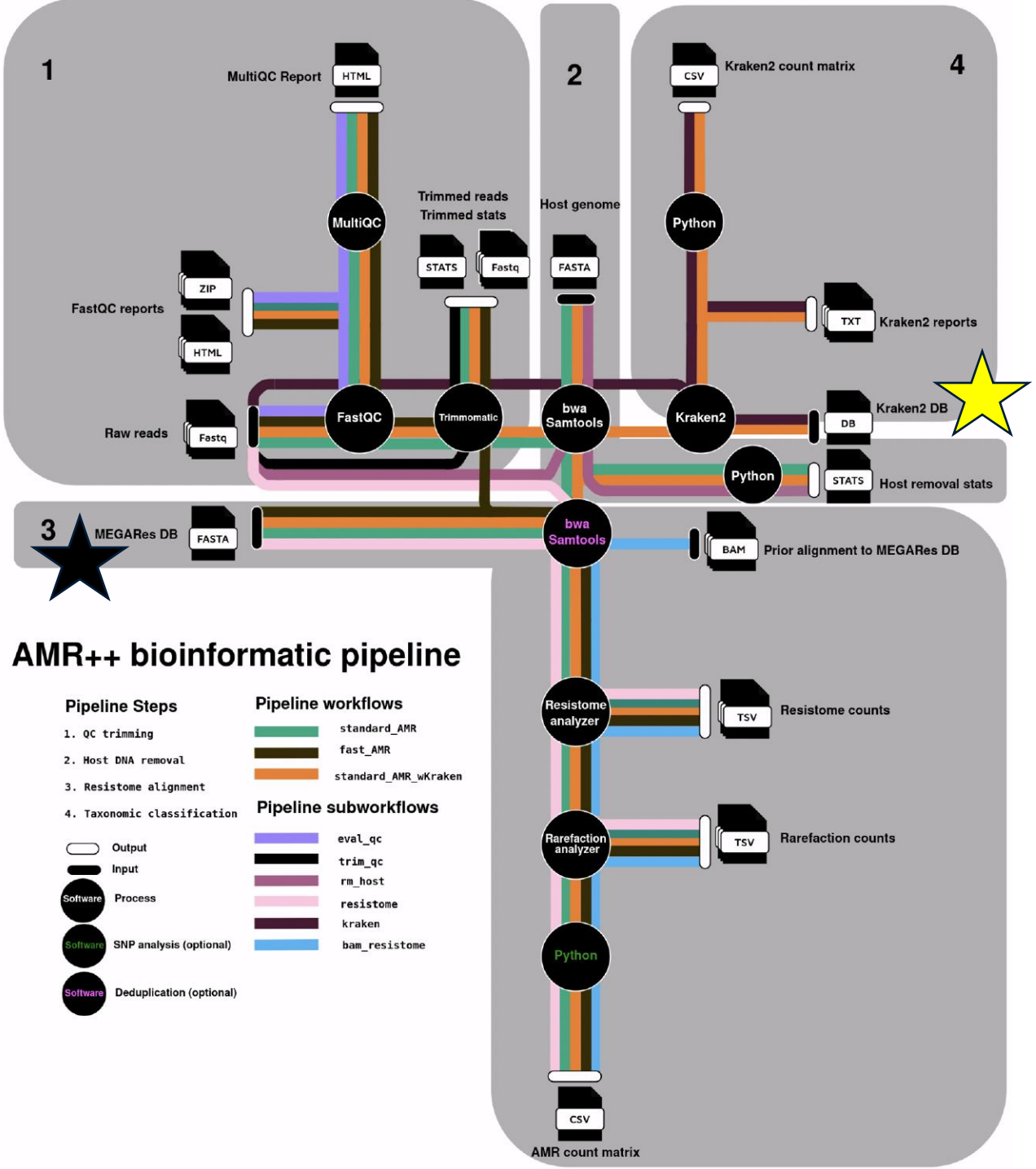
- http://jnci.oxfordjournals.org/content/92/16/1295.full

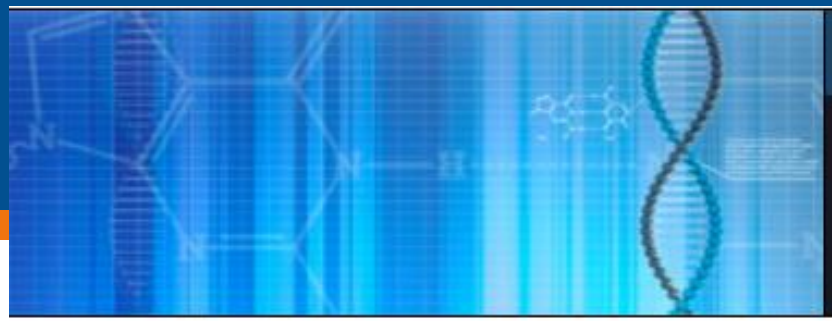
Mechanisms

- MDR 23S ribosomal RNA methyltransferase
- MDR 23S rRNA mutation
- MDR acetyltransferase
- MDR mutant porin proteins
- MDR regulator
- Multi-drug ABC efflux pumps

>>MEG_2694|Drugs|Multi-drug_resistance|Multi-drug_ABC_efflux_pumps|EATAV|RequiresSNPConfirmation

ATGTCTAAAATCGAAATAAAAAATCTGACATTCGGCTACGACAGCCAAGGCACATTATTATTTGAACAAGCAAATCTAAATTTTGACACACAATGGAAACTAG
GACTTATCGGACGAAACGGTCGAGGAAAGACAACTTTACTGAATATTCTACAAAACAAACTACCTTATCAAGGGCAAGTAATCCATCAGCAAGAATTTGCCTA
TTTCCCGCAACAGACAAAAGATAAAGAACGTTTAACCTATTACGTGTTAAATGATATTACGGATTTTGAGATATGGGAAATCGAAAGAGAGCTCCAATTGATG
CAAACAGATCCTGAAATCTTATGGAGAGAATTCAGCACACTATCGGGGGGAGAGAAGACAAAAGTCCTACTGGCACTTTTATTTGTGGATGACACTCATTTCC
CGTTAATCGATGAACCAACGAATCATTTGGATATCTCTGGTAGAAAACAAGTAGCGGCTTATTTGAAAAAGAAAAAACAAGGCTTCATCGTGGTCAGCCATGA
CCGGGGATTTATCGATGAAGTAGTGGACCATGTTTTAGCAATCGAAAAAAGTCAACTGGAACTTTATCAAGGGAATTTCTCTATCTATGAAGAACAGAAAAAA
CTTCGTGATGAATTTGAAATGGCTCAAAATGAAAAATTGAAAAAAGAAGTCAGTAGGCTAAAGAAAACAGCAGCTGAAAAAGCCGAATGGTCTCGTTCCCGAG
AAGGAGATAAAACAAAGAAACAAGTCGGATTCATCGATACTGAATCTAGACGAGTGAATAAAGGAGCAGTGGGTGCTGATGCTGCACGGACGATGAAACGATC
CAAAGCAATCGTGAATCGGATGGAGACCCAGATCAGCGAGAAAGAAAAACTATTAAAAGATATCGAATATATCGATTCTTTGACGATGAATAGCCAAGCGTCT
CACCATAAGCGACTTTTAAGCGTAGAAGATCTTCAATTAGGGTATGAAAATCTGTTATTCGAGCCAATTCATTTTACAATCGAGCCTCATCAGCGGGTGGCGA
TTTCAGGTCCTAACGGTGCAGGAAAGTCATCCATTATCCATTATCTTCTGGGGGCATTCAACGGCAAGGTTATAGGAGAAAAATACCAGCCAAAACATCTGAG
CATTAGTTATGCAAGCCAAAATTATGAAGACAATCGAGGAACGTTGGCGGAATTTGCAGAGAAAAACCAAGTAGACTACCAAGCATTTTTGAACAACCTCCGA
AAGCTTGGGATGGAAAGAGATGTTTTTCATAACAAGATCGAGCAGATGAGTATGGGCCAACGGAAAAAAGTGGAATTGGCTAAATCTTTATCACAGCCAGCTG
AACTATATATATGGGATGAACCATTGAATTATTTGGATGTCTTCAATCAAGAACAATTAGAACAACTGATCTTGAACGTGAAACCTGCCATGTTACTAGTGGA
ACATGATCAAACCTTCCTGGATAAAGTATCTACTGAGATTATTTCTCTTGAGAGAATCTAA


1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome



RefSeq: NCBI Reference Sequence Database

A comprehensive, integrated, non-redundant, well-annotated set of reference sequences including genomic, transcript, and protein.

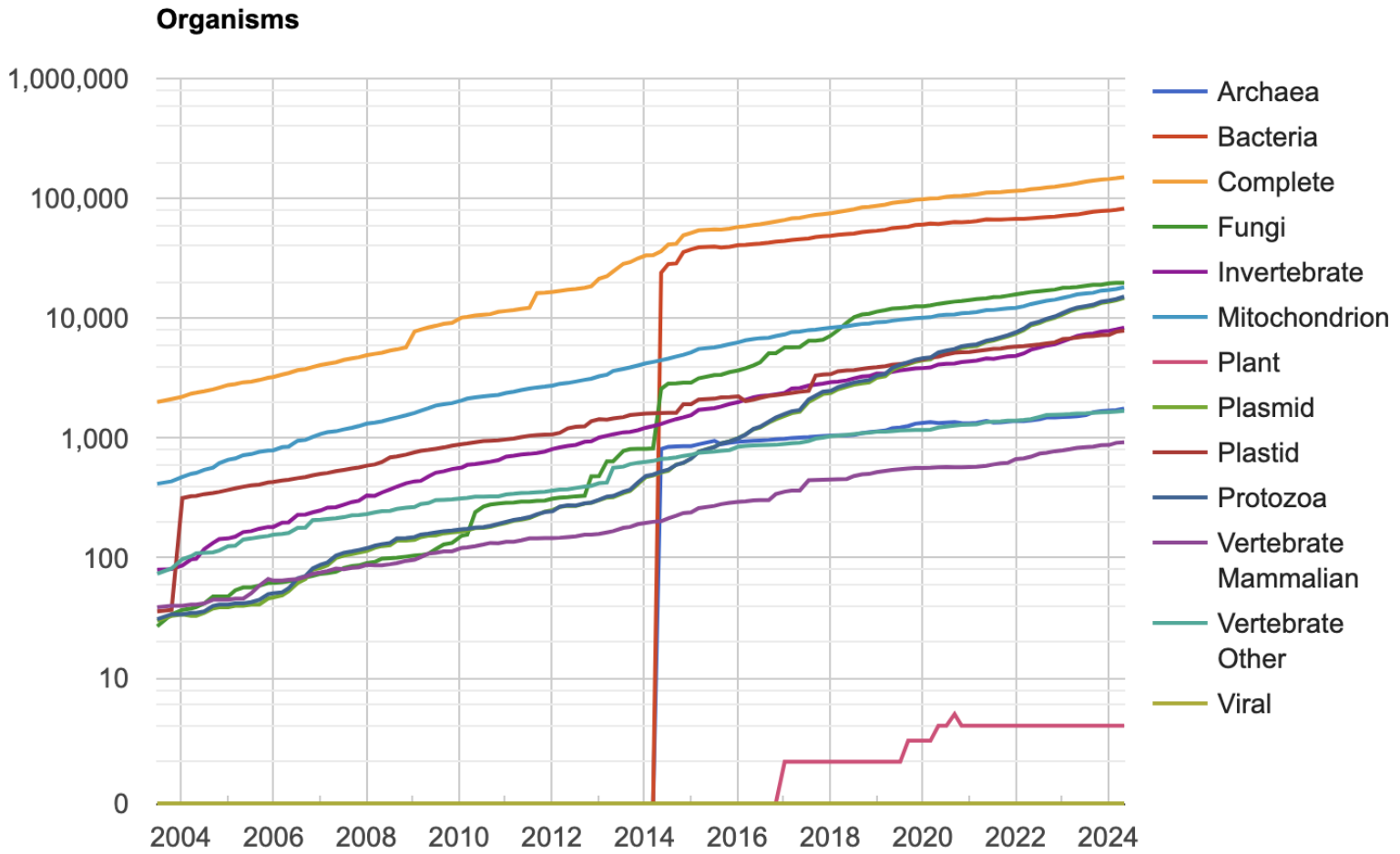
"Comprehensive" – it's relative

"Integrated" -- ??

"Non-redundant" -- ??

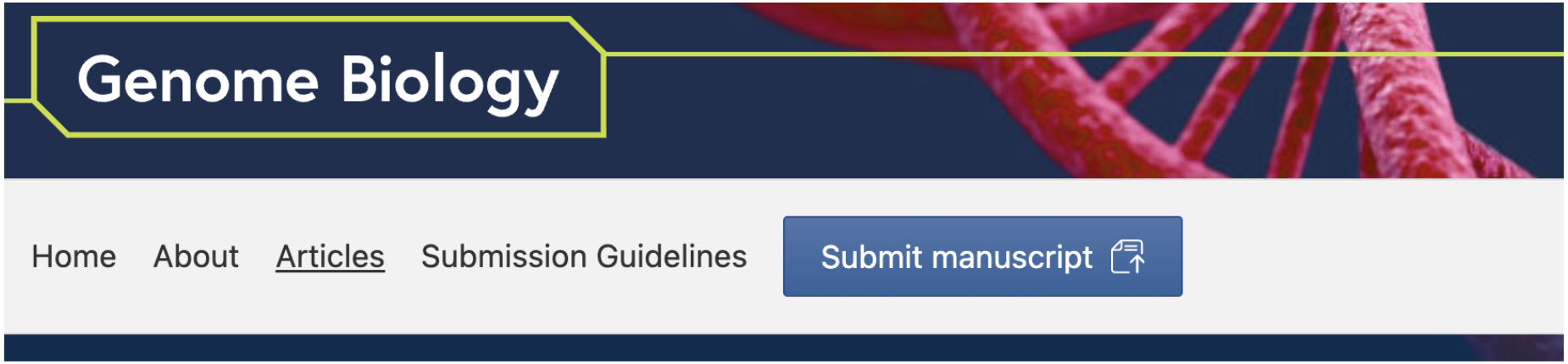
"Well-annotated" -- not really

RefSeq Bacteria Growth Statistics



https://www.ncbi.nlm.nih.gov/refseq/statistics/

Database size does matter, and there's good research on this:



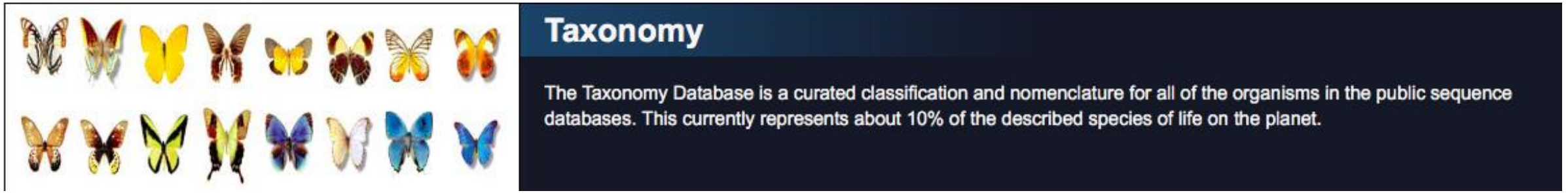
Open Letter | Open access | Published: 30 October 2018

RefSeq database growth influences the accuracy of *k*-mer-based lowest common ancestor species identification

Daniel J. Nasko, Sergey Koren, Adam M. Phillippy & Todd J. Treangen

Genome Biology **19**, Article number: 165 (2018) | Cite this article

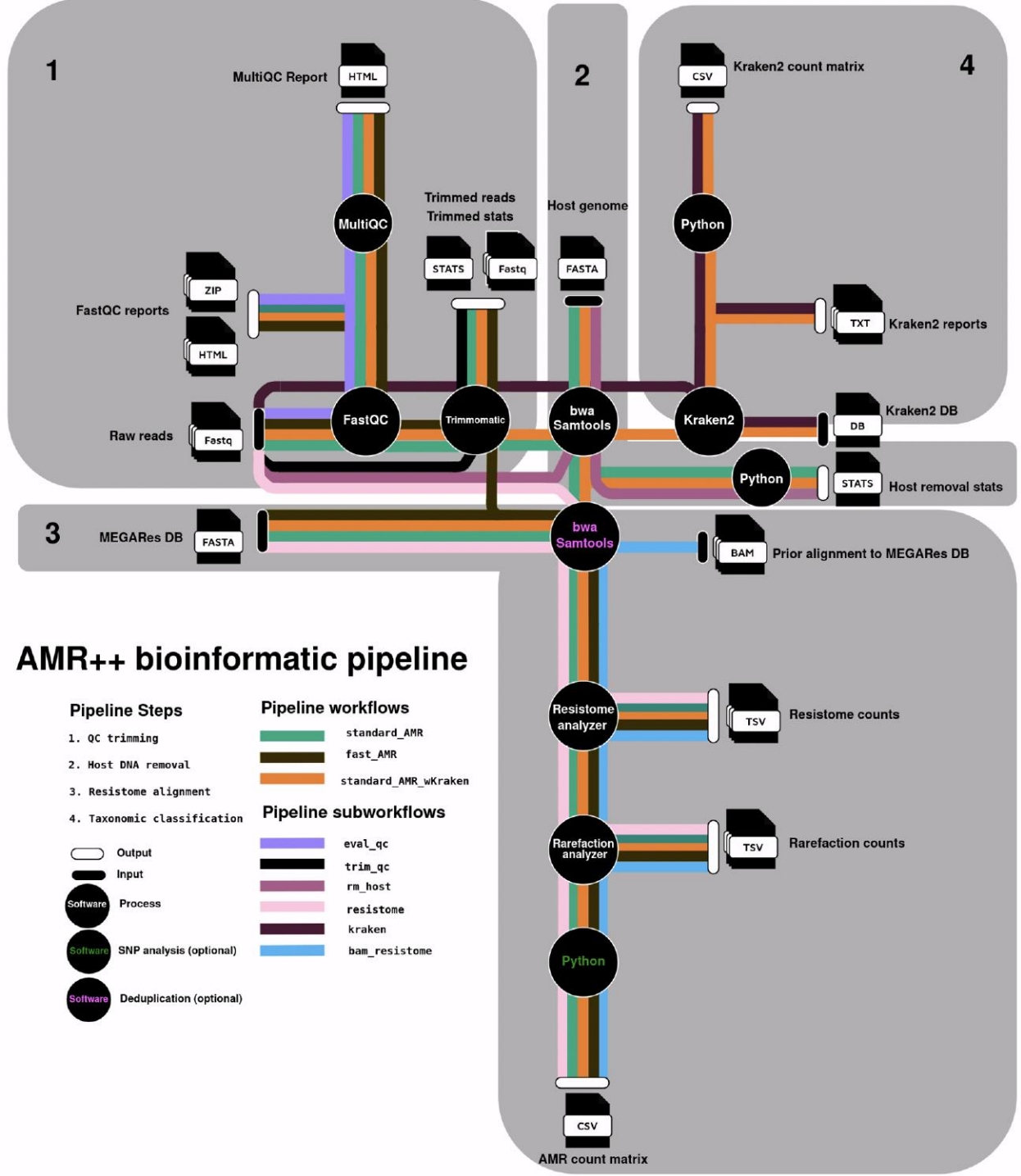
The NCBI Taxonomy Database



Also has its limitations.

In the end, keep in mind that taxonomies are attempting to bin DNA sequences into containers with “hard boundaries”.

Part 2: Quality assessment of shotgun metagenomic data


1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome

Why do we need to do quality assessment?

1. Because sequence data can have errors – both expected and unexpected
2. Because sequence data usually contain both technical and biological sequence

Example of adapter read-through

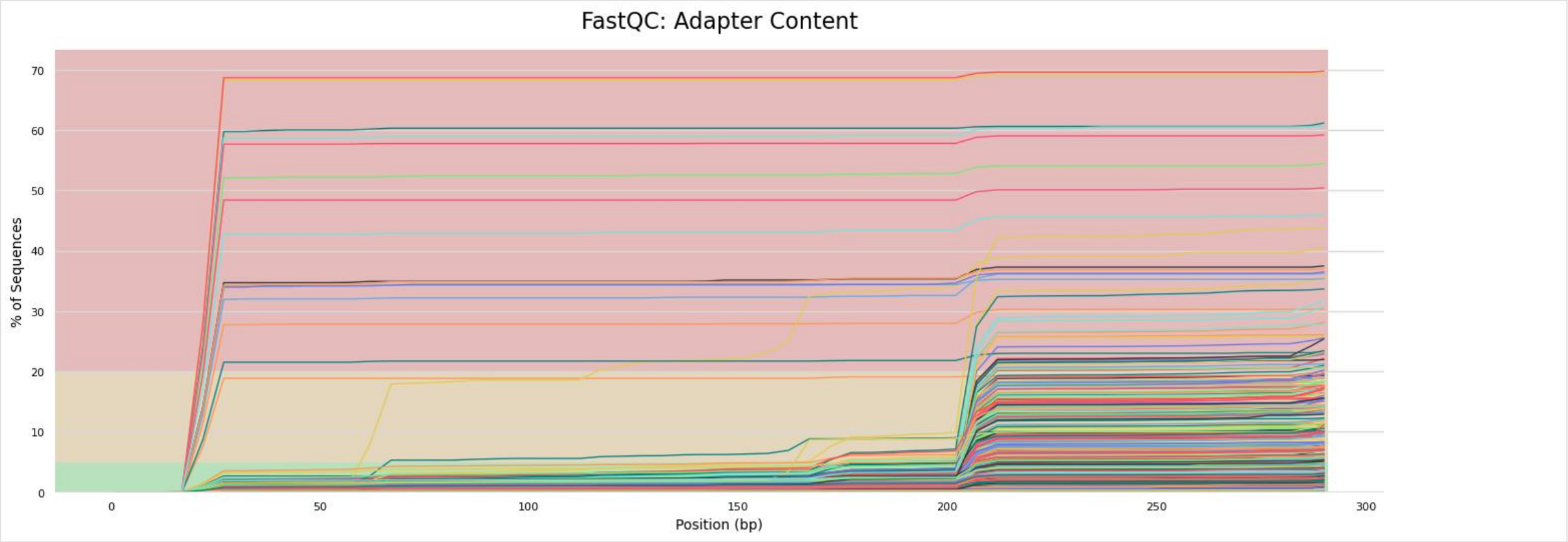
Adapter Content

72 47 115

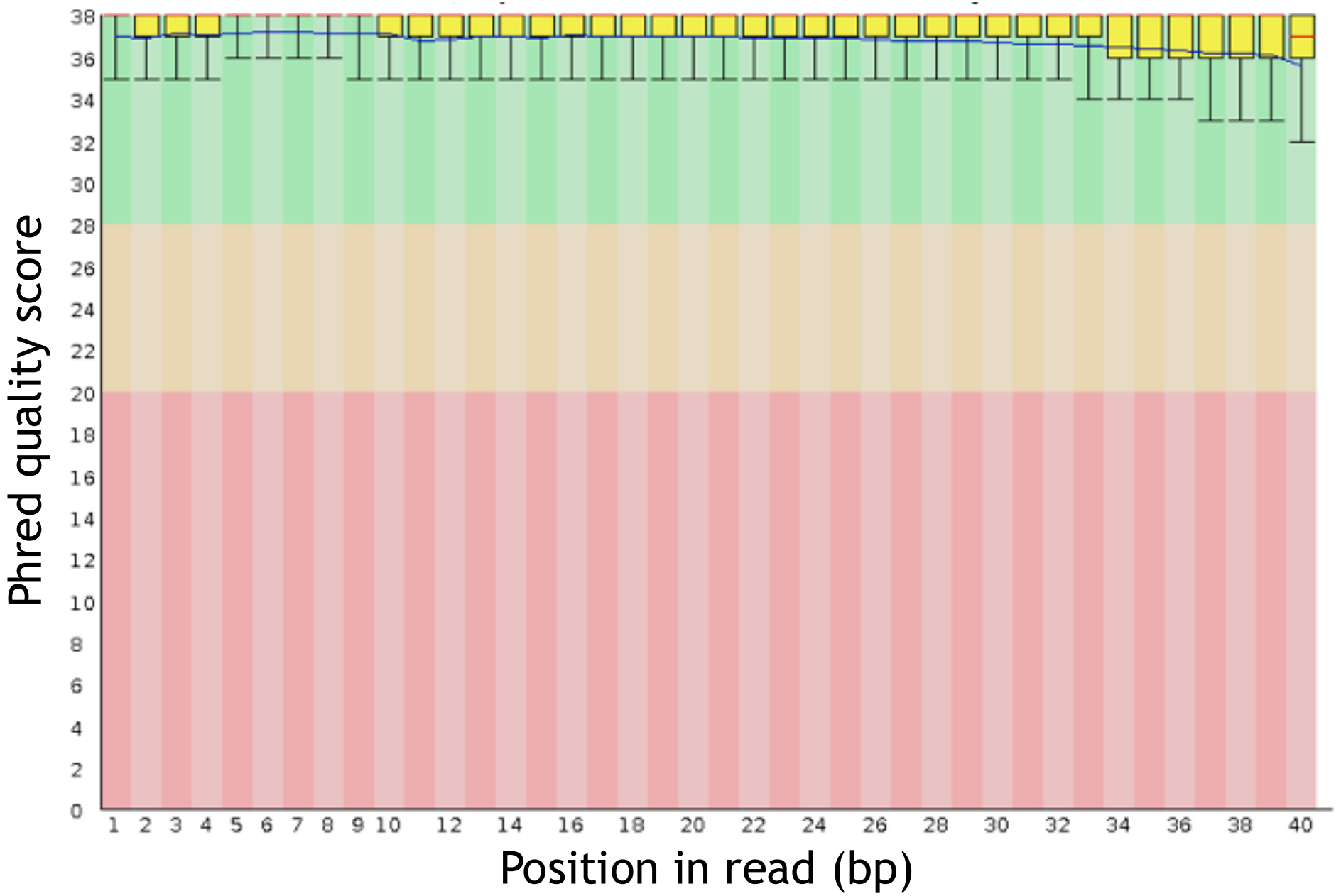
Help

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

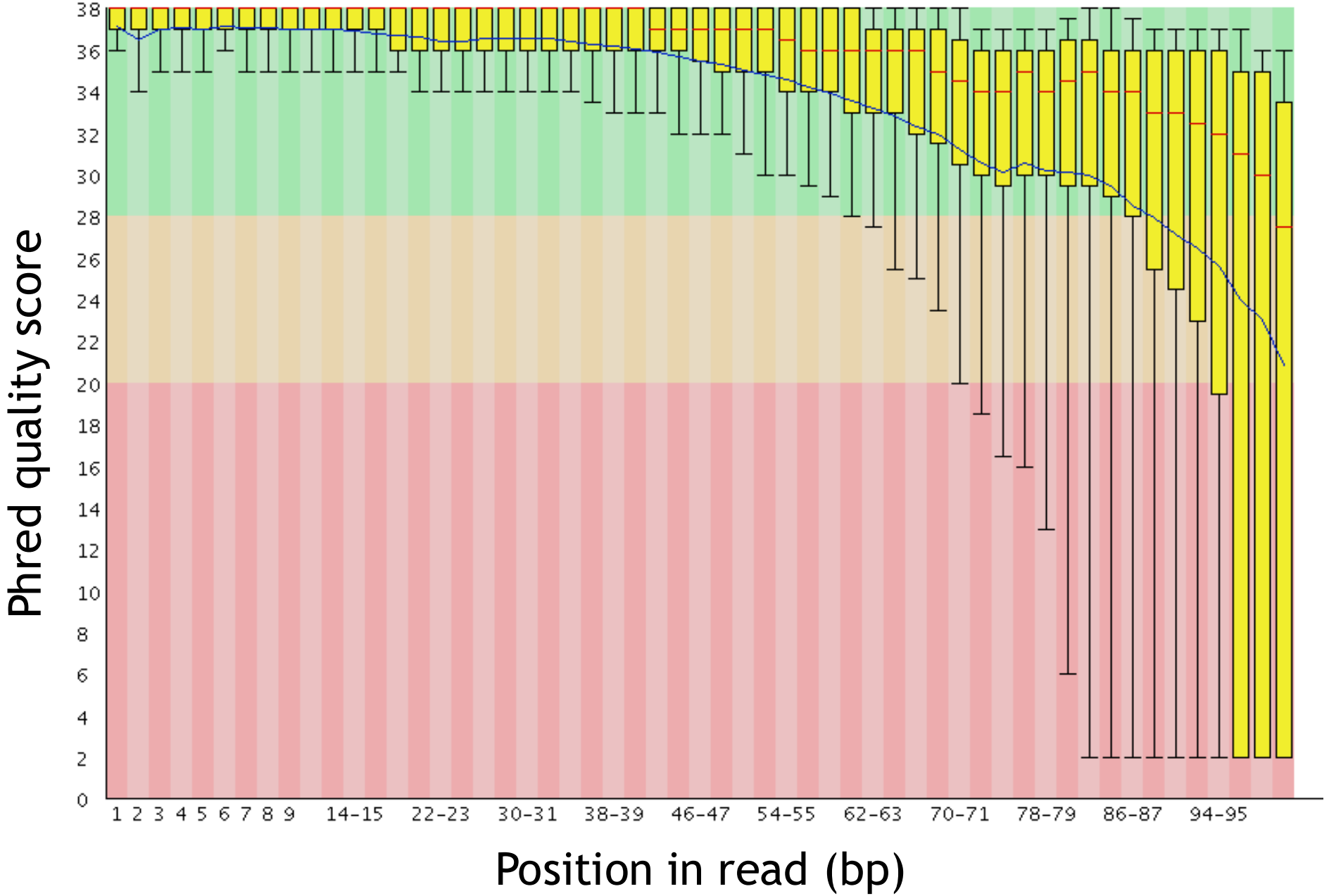
Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).



FastQC Output: Good Illumina Data



FastQC Output: Bad Illumina Data



Trimming of FASTQ Sequences

Trimming corrects several error types:

- Illumina systematic bias (end of read base substitutions)
- Adapter sequences
- Missing mate-pairs
- Low quality reads

Filtering of Off-Target Reads

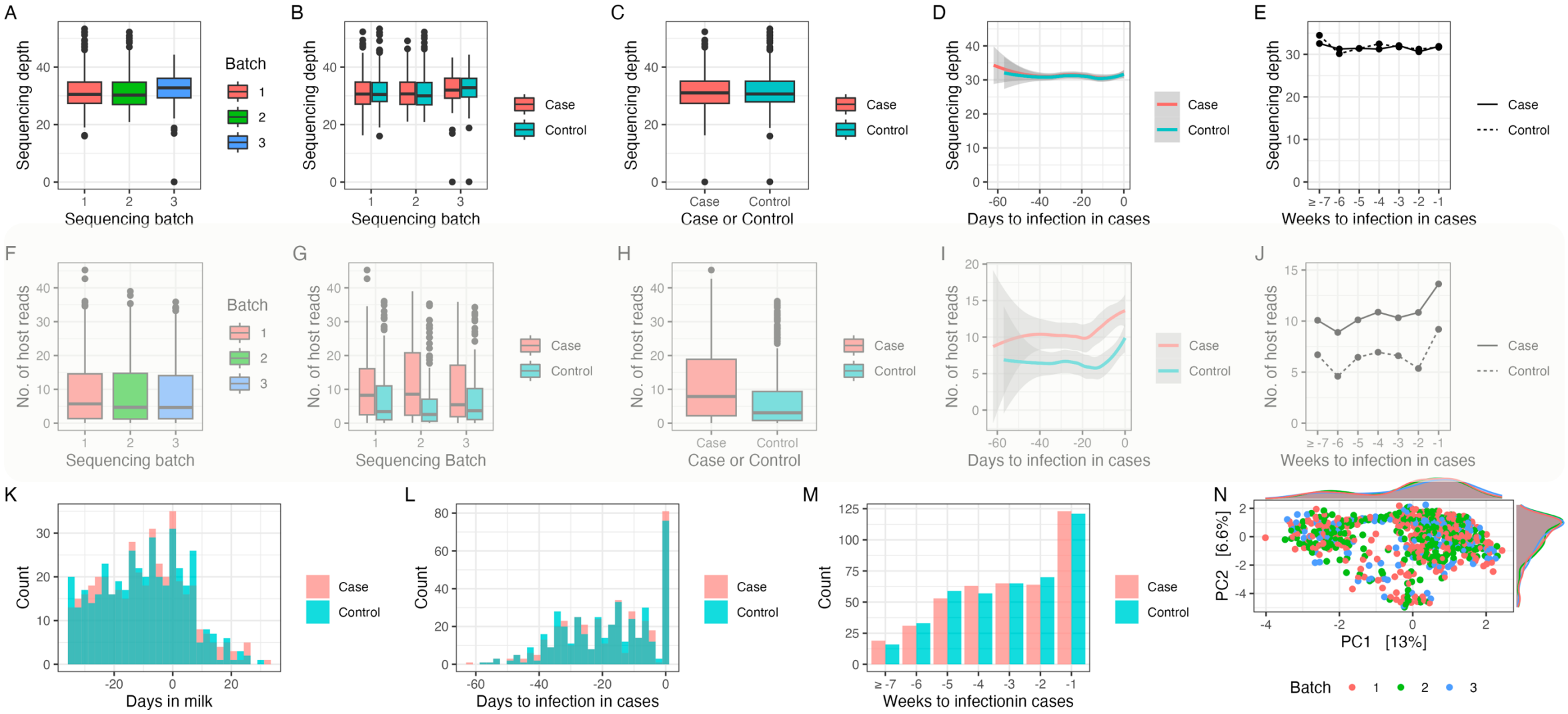
- Samples collected from hosts contain host DNA
- Known contaminants might be present due to library preparation techniques

Filtering of Off-Target Reads

- Filtering host DNA removes the chance of false positive classification

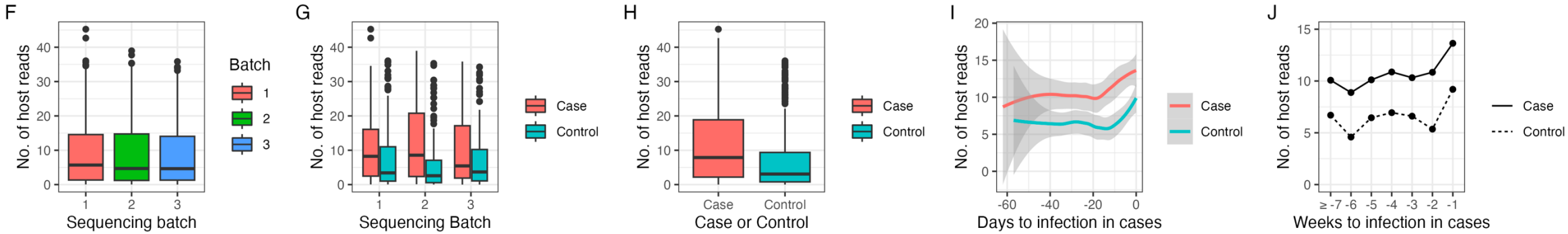
- AMR++ filtering:
  - Single filtering genome
  - Multiple filtering genome

This can uncover some interesting findings!



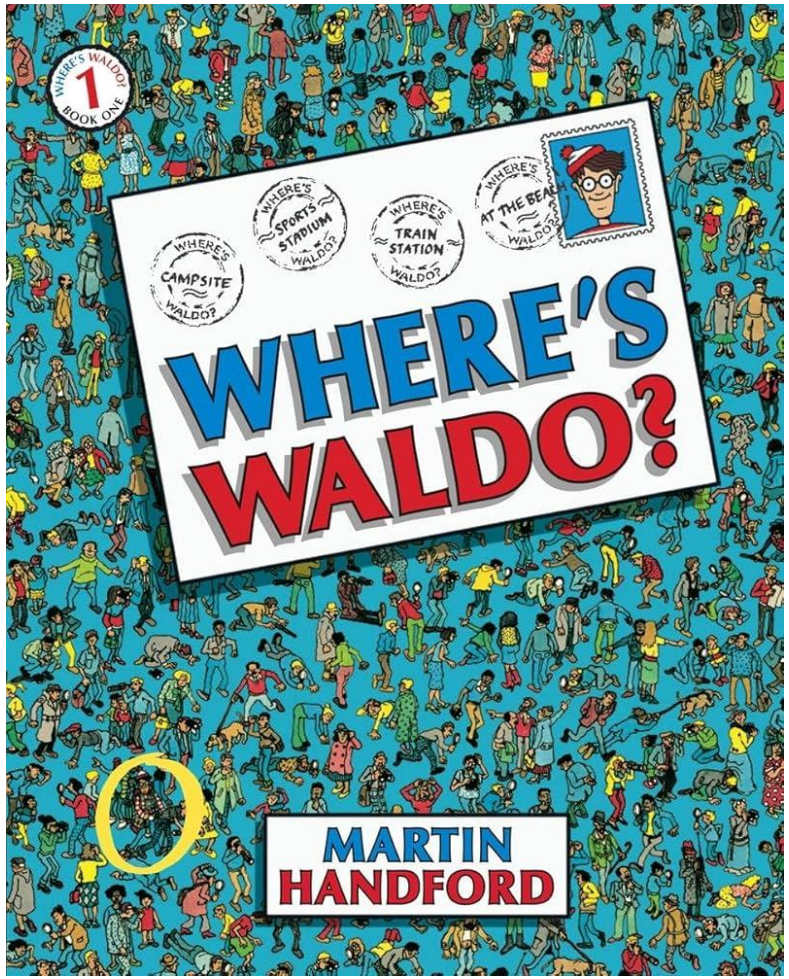
Dean CJ, PeñaMosca F, Ray T, Wehri TJ, Sharpe K, Antunes AM, Doster E, Fernandes L, Calles VF, Bauman C, Godden S, Heins B, Pinedo P, Machado VS, Caixeta LS, Noyes NR. 2024. Exploring associations between the teat apex metagenome and *Staphylococcus aureus* intramammary infections in primiparous cows under organic directives. Appl Environ Microbiol 90:e02234-23.https://doi.org/10.1128/aem.02234-23

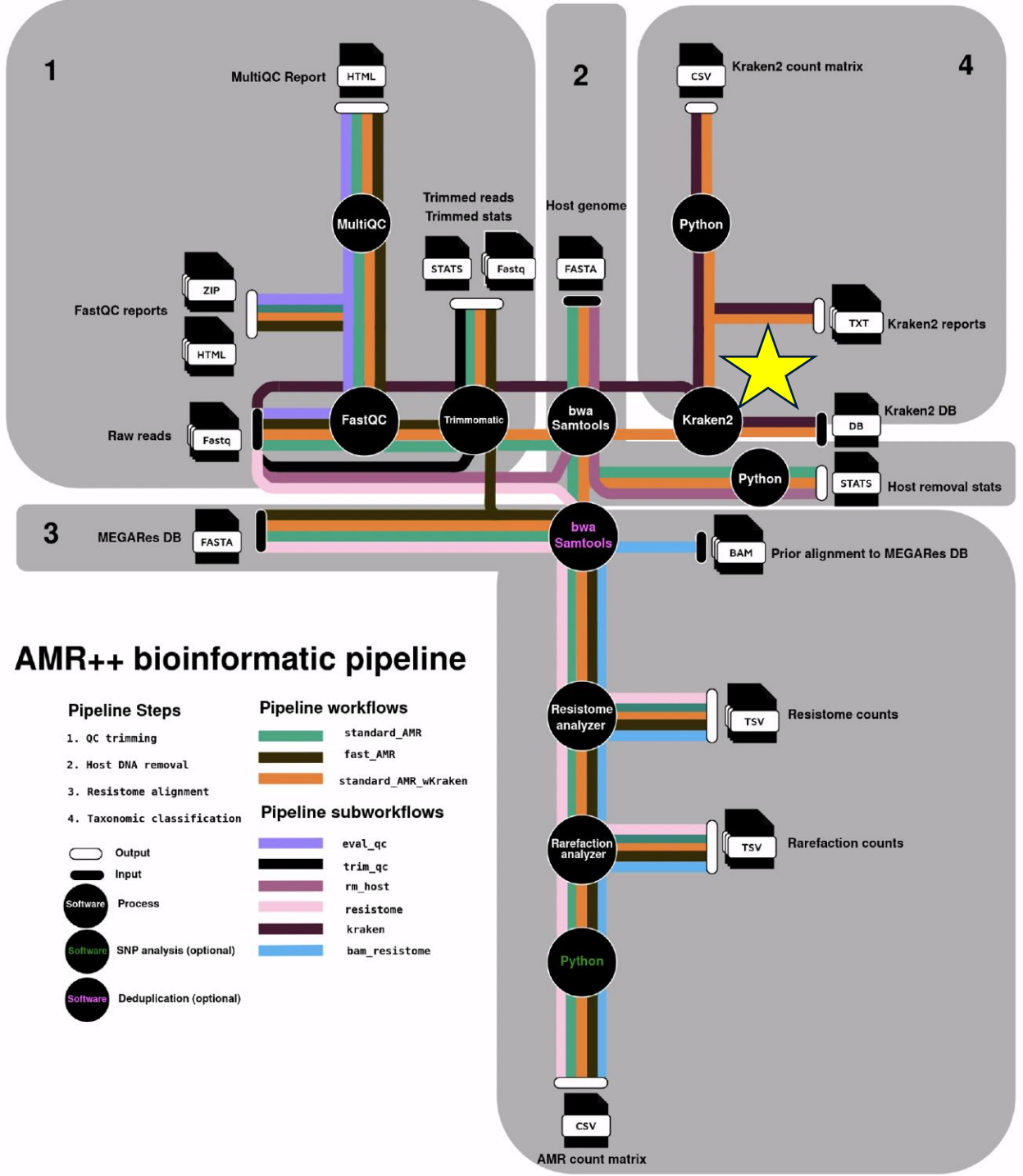
This can uncover some interesting findings!



Dean CJ, PeñaMosca F, Ray T, Wehri TJ, Sharpe K, Antunes AM, Doster E, Fernandes L, Calles VF, Bauman C, Godden S, Heins B, Pinedo P, Machado VS, Caixeta LS, Noyes NR. 2024. Exploring associations between the teat apex metagenome and *Staphylococcus aureus* intramammary infections in primiparous cows under organic directives. Appl Environ Microbiol 90:e02234-23.https://doi.org/10.1128/aem.02234-23

Part 3: *WHO* is there?




1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome

Let's assume that we want to create a simple microbiome profile for 1 sample

Let’s assume that we want to create a simple microbiome profile for 1 sample

Basically, we want to know:

WHO is there?
What is their ***RELATIVE ABUNDANCE***?

Let's start with *WHO* is there…

Two approaches for determining who:

1. Marker-based
2. Genome-based

Marker-based methods = “16S on steroids”

Examples:

MetaPhlAn

PhyloSift

And many others…

Two approaches for determining who:

1. Marker-based

2. Genome-based

How could the metagenome composition be used to bin and classify sequence data?

- GC-content, nucleotide frequencies, codon patterns

- *k*-mer based
    a. exact-match
    b. profile

Each Sequence has a k-mer Profile

For every sequence, there exists a profile of k-mers

e.g., Sequence: ACGTGTG

3-mer profile:

```
ACG: 1
CGT: 1
GTG: 2
TGT: 1
```

k-mer Profiles Contain Information

- *k*-mer profiles can help to identify a sequence
- Match the *k*-mer profile of unknown sequences to known *k*-mer profiles
- This is a classification problem:
  - We have an unknown observed sequence
  - We have known genomes
  - We can use matching of *k*-mer profiles to label the unknown sequence

What is Kraken?

Kraken is actually a program that tries to match your sequence data to all of the genomes in RefSeq *(more on this later)*…

So really, it's the NCBI RefSeq database, "manipulated" in a special way

What is Kraken?

Kraken pulls all of the prokaryotic and viral genomes from RefSeq and does some algorithmic maneuvering to create the “Kraken-DB”

Then, it uses the NCBI taxonomy to name everything.

What is Kraken?

- This is kraken's "database"; it is a database of *k*-mer profiles created from NCBI RefSeq genomes, which are also linked to NCBI's taxonomy

- This database is large, and it requires a lot of computational resources to build it; however, once it is built, it can "sit" on your computer to be used over and over again (can also be updated periodically)

How is it created (high level)?

1. Chop up all of the genomes in RefSeq
2. Compare all of the pieces to one another
    a. Pieces ***without*** a match represent the RefSeq genome ”as-is” (strain, species, genus, etc.…)
    b. Pieces ***with*** match(es) get moved up the taxonomy until they are unique within a “level” (the “lowest common ancestor” or LCA approach)

Let’s do an example….

After chopping up all RefSeq genomes into 4-mers, we have identified that the 4-mers `ACCT` and `GGTA` are contained within these genomes, and we want to uniquely associate them with taxa in the NCBI taxonomy.

Kingdom

Phylum

Class

Order

Family

Genus

Species

Which species is it?

Salmonella enterica ssp. arizonae

Salmonella enterica ssp. enterica

Salmonella bongori

Escherichia coli

Kingdom

Phylum

Class

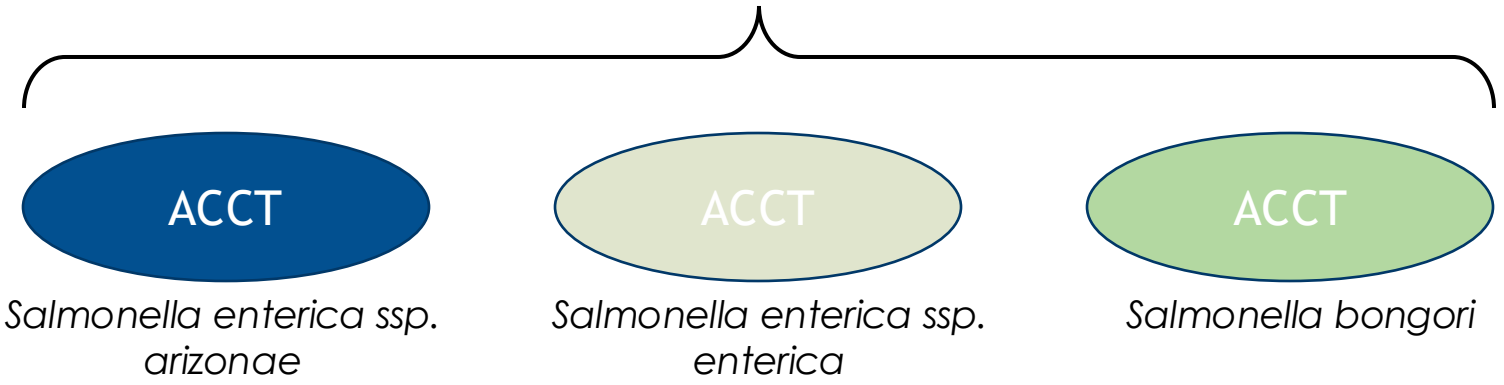
Order

Family

Genus

Species

How could we resolve this?



Kingdom

Phylum

Class

Order

Family

Genus

Species

Salmonella *Salmonella* *Salmonella*

Kingdom

Phylum

Class

Order

Family

Genus

Species

Uh-oh…..

Salmonella *Salmonella* *Salmonella* *Yersinia*

Kingdom

Phylum

Class

Order

Family

Genus

Species

Enterobacteriaceae

Enterobacteriaceae

Enterobacteriaceae

Enterobacteriaceae

What do you end up with? The database…

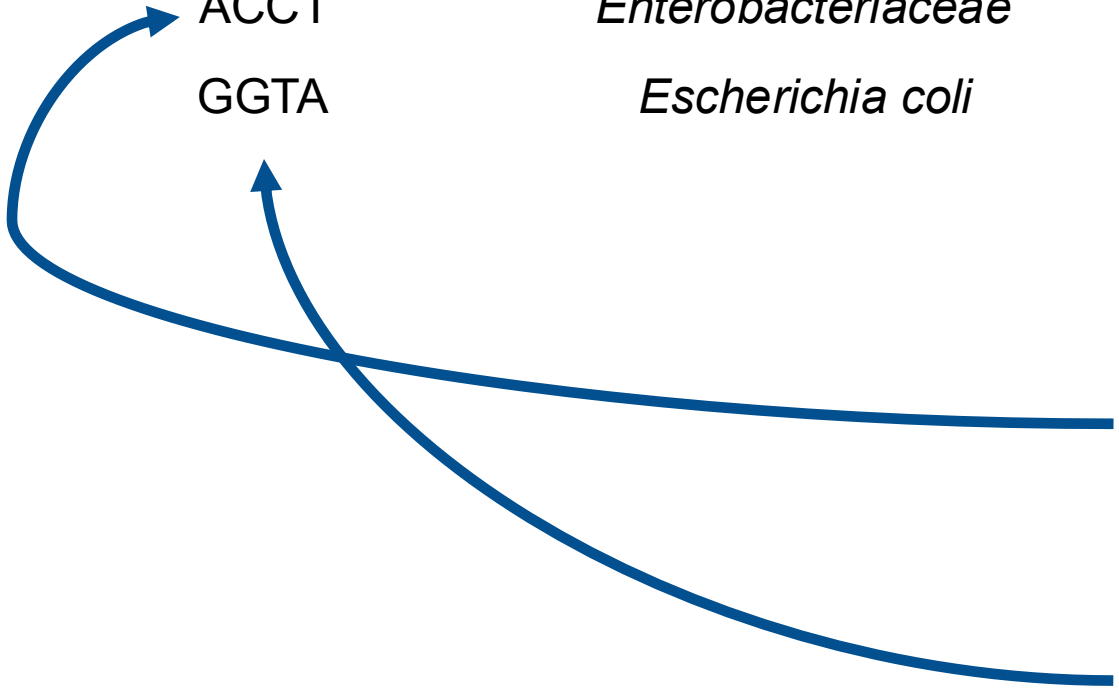
***K*-mer sequence**	**NCBI taxonomic ID**	**kraken ID**
ACCT	*Enterobacteriaceae*	1339265
GGTA	*Escherichia coli*	3927561

How do we use this database to classify the reads in our metagenomic data?

***K*-mer sequence**	**NCBI taxonomic ID**	**kraken ID**
ACCT	*Enterobacteriaceae*	1339265
GGTA	*Escherichia coli*	3927561

When you then run your sequence data through kraken, it looks for exact *k*-mer matches

***K*-mer sequence**	**NCBI taxonomic ID**	**kraken ID**
ACCT	*Enterobacteriaceae*	1339265
GGTA	*Escherichia coli*	3927561

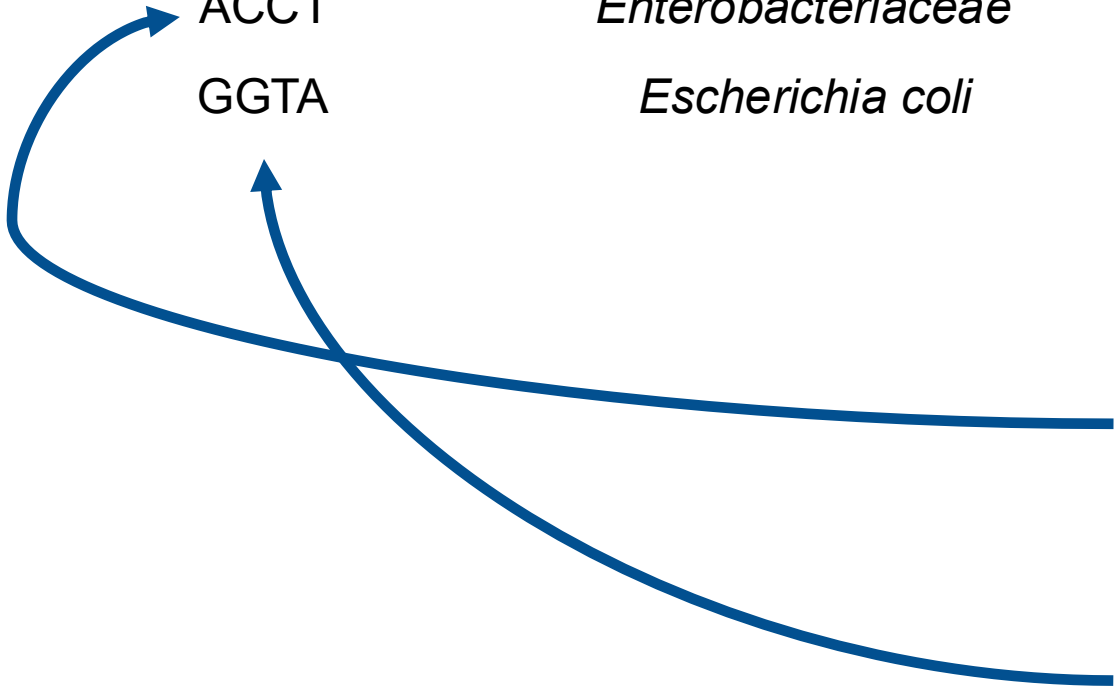


Your Fastq:

HiSeq2000:123456
ACCT
+
#!%&
HiSeq2000:123457
GGTA
+
HHII

When you then run your sequence data through kraken, it looks for exact *k*-mer matches

***K*-mer sequence**	**NCBI taxonomic ID**	**kraken ID**
ACCT	*Enterobacteriaceae*	1339265
GGTA	*Escherichia coli*	3927561



Your Fastq:

HiSeq2000:123456
ACCT
+
#!%&
HiSeq2000:123457
GGTA
+
HHII

→ 1 count for *Enterobacteriaceae*

→ 1 count for *Escherichia coli*

But sometimes, it is messy…

- Kraken's default k-mer size for the exact-matching process is 31

- Kraken takes every 31-mer from every read and attempts to match it (exactly!) to the kraken database
  - Some 31-mers don't match (exactly) to anything = unclassified
  - A single read is usually 150-300bp, and therefore it contains many 31-mers. Sometimes, those 31-mers match to *different* things. In that case…there is a confidence scoring system.

Benefits and Limitations (some)

The Good:

- Rapid
- High specificity (**caveat: for those organisms in the RefSeq database)
- Pulls from a relatively well-curated database (**caveat: plasmids)

The Bad:

- High rate of “unclassified”
- Varying “degrees” of taxonomic resolution (and probably non-random)
- Pulls from a relatively small database

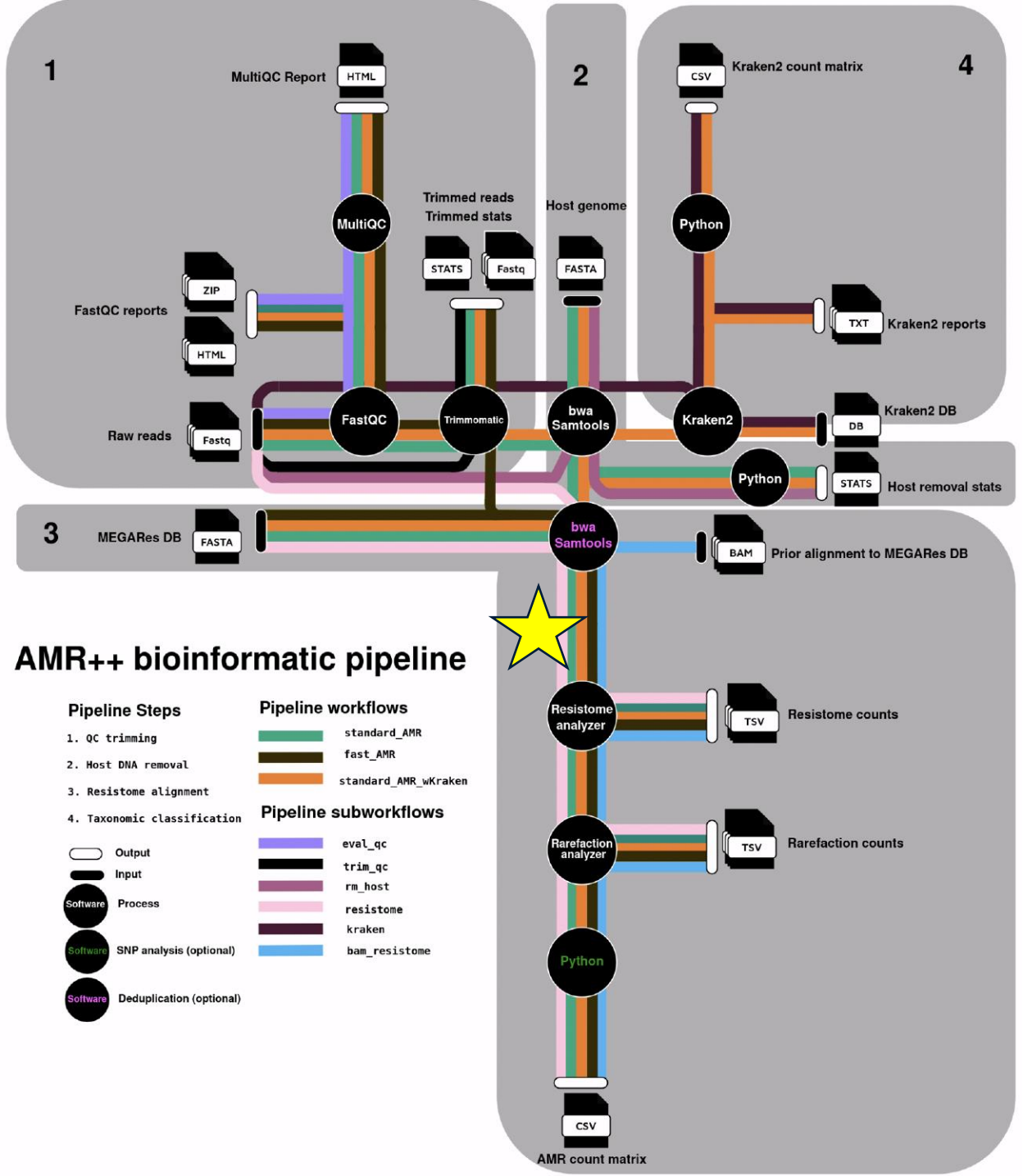
More details (from user manual):

OUTPUT: **`562:13 561:4 A:31 0:1 562:3`** would indicate that:

- the first 13 *k*-mers mapped to taxonomy ID #562
- the next 4 *k*-mers mapped to taxonomy ID #561
- the next 31 *k*-mers contained an ambiguous nucleotide
- the next *k*-mer was not in the database
- the last 3 *k*-mers mapped to taxonomy ID #562

In this case, ID #561 is the parent node of #562. Here, a label of #562 for this sequence would have a score of C/Q = (13+3)/(13+4+1+3) = 16/21. A label of #561 would have a score of C/Q = (13+4+3)/(13+4+1+3) = 20/21. If a user specified a confidence threshold over 16/21, the classifier would adjust the original label from #562 to #561; if the threshold was greater than 20/21, the sequence would become unclassified.

Part 4: AMR Detection


1
MultiQC Report
HTML
MultiQC
Trimmed reads
Trimmed stats
STATS
Fastq
FastQC reports
ZIP
HTML
Raw reads
Fastq
FastQC
Trimmomatic
2
Host genome
FASTA
bwa Samtools
4
CSV
Kraken2 count matrix
Python
TXT
Kraken2 reports
Kraken2
DB
Kraken2 DB
Python
STATS
Host removal stats
3
MEGARes DB
FASTA
bwa Samtools
BAM
Prior alignment to MEGARes DB
AMR++ bioinformatic pipeline
Pipeline Steps
1. QC trimming
2. Host DNA removal
3. Resistome alignment
4. Taxonomic classification
Output
Input
Software
Process
Software
SNP analysis (optional)
Software
Deduplication (optional)
Pipeline workflows
standard_AMR
fast_AMR
standard_AMR_wKraken
Pipeline subworkflows
eval_qc
trim_qc
rm_host
resistome
kraken
bam_resistome
Resistome analyzer
TSV
Resistome counts
Rarefaction analyzer
TSV
Rarefaction counts
Python
CSV
AMR count matrix

What is our goal?

To describe and/or compare resistomes in a set of samples!!

To do that, we must identify and quantify AMR genes

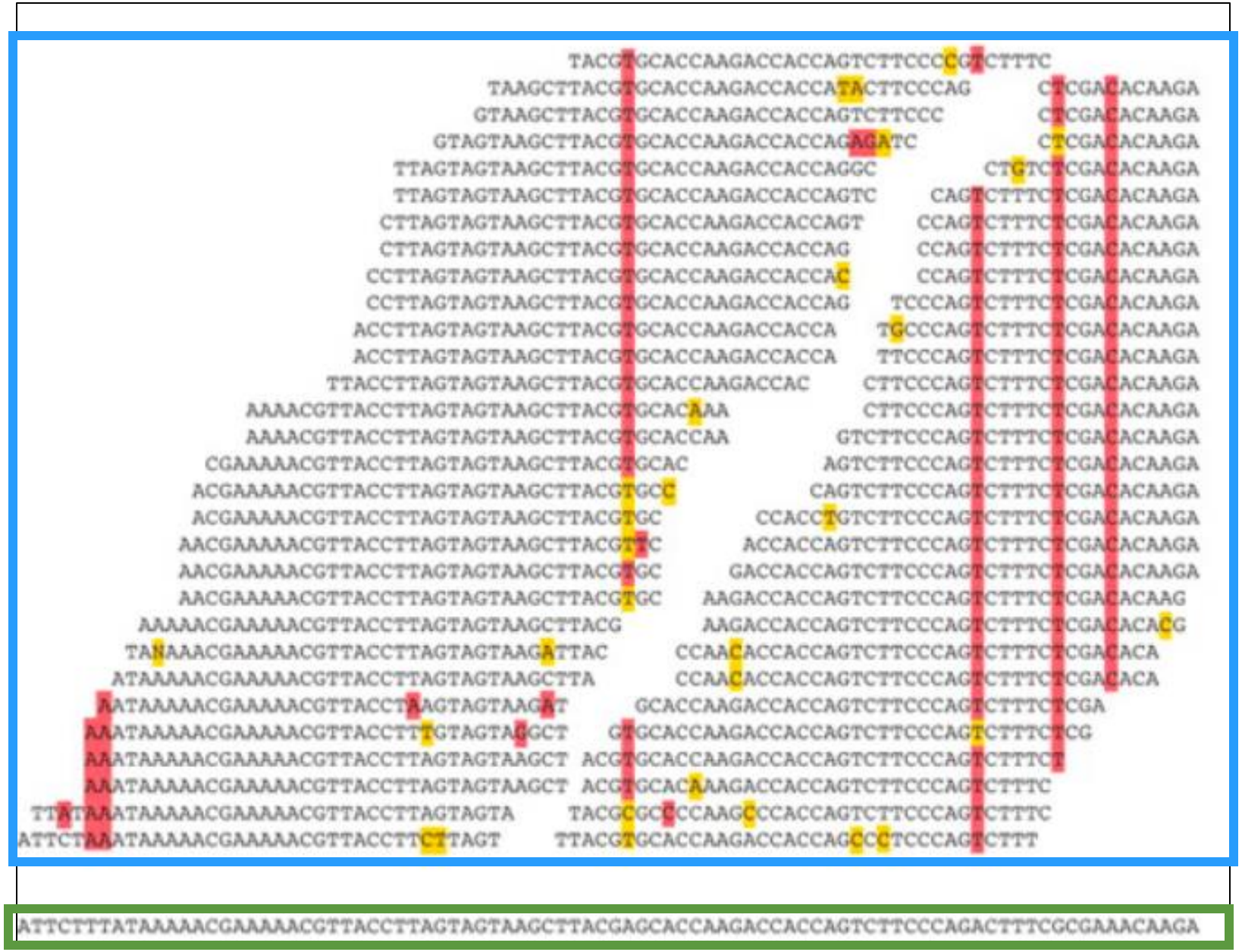
To do that, we must find and count AMR genes in metagenomic data

Remember, that for all of the options we will cover today, it all comes down to matching…

(And that's why the databases are so important!)

Basic Alignment Approach

Sequence reads



Reference Sequence

Important characteristics:

- The match does not have to be perfect
- Matching is performed at the unit of the read
- The alignment preserves a lot of information

Benefits of alignment:

1. Utilizes all of the (relevant) sequence data
2. Relatively quick (computationally)
3. Preserves sequence variability
4. Preserves the original sequence (no “assembly error”)

Disadvantages of alignment:

1. Does not provide genomic context
2. Can be prone to false-positive identification of, e.g., AMR genes
3. Must make arbitrary decisions when there are homologous stretches of DNA between reference sequences

Disadvantages of alignment:

1. Does not provide genomic context
2. Can be prone to false-positive identification of, e.g., AMR genes
3. Must make arbitrary decisions when there are homologous stretches of DNA between reference sequences

Some examples:

Some examples:

Some examples:

Some examples:

So what do we do about this?

We use a “gene fraction” cutoff

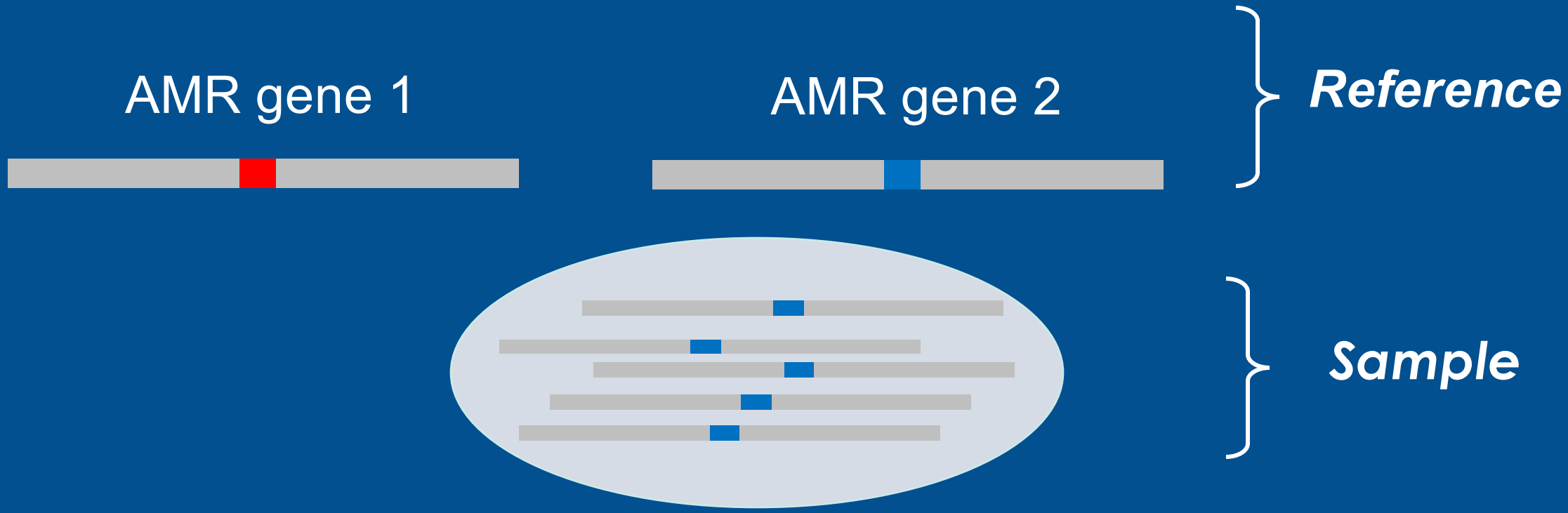
(sometimes called gene “coverage”: AVOID THIS TERM)

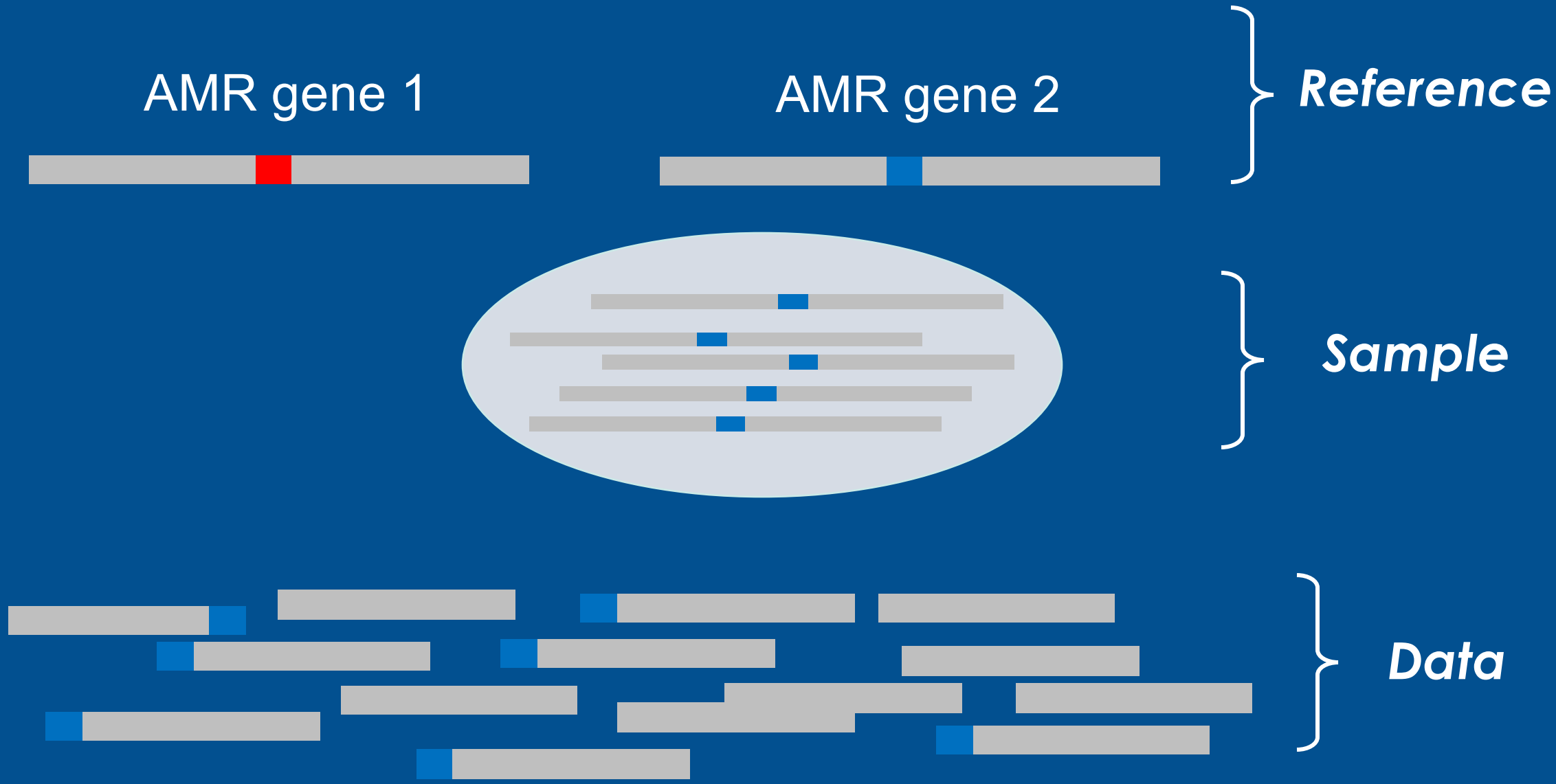
Gene Fraction: The % of nucleotides in a given reference sequence that have at least one mapped (aligned) read

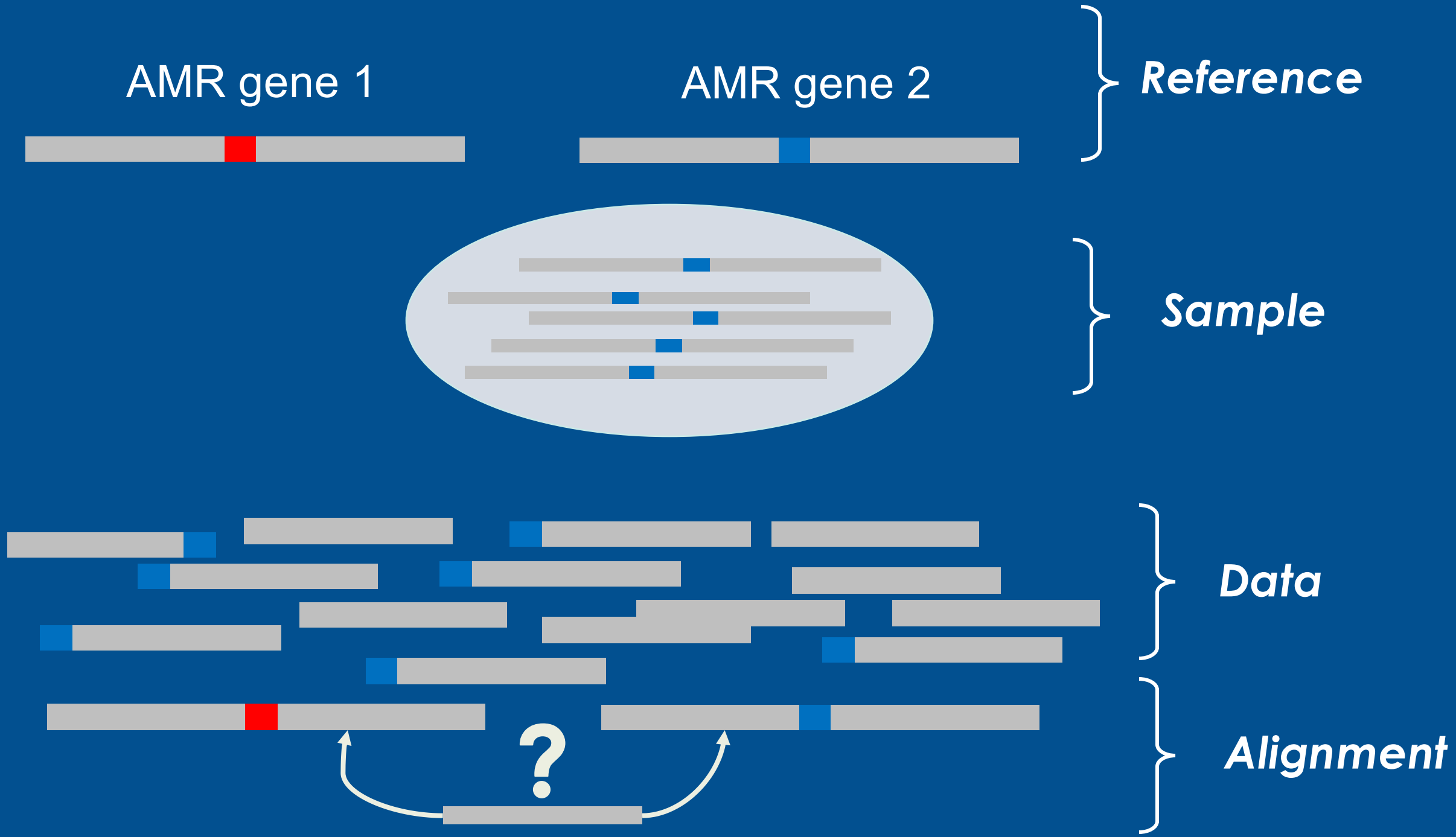
Disadvantages of alignment:

1. Does not provide genomic context
2. Can be prone to false-positive identification of, e.g., AMR genes
3. Must make arbitrary decisions when there are homologous stretches of DNA between reference sequences


AMR gene 1
AMR gene 2
Reference


AMR gene 1
AMR gene 2
Reference
Sample


AMR gene 1
AMR gene 2
Reference
Sample
Data


AMR gene 1
AMR gene 2
Reference
Sample
Data
Alignment
?

What can we do?

Default of the aligner = random assignment

You can also force aligner to output ***all possible alignments*** (but this means you will "double count")

You can "sidestep" the issue by conducting your analyses at a "higher level" – more on this later!