

# ESGEM-AMR



[amrrules.org](http://amrrules.org)

# Agenda

- 1. Online resources:** where to find things
- 2. AMRrules Python package:** applying rules to interpret genotypes
- 3. AMRrulemakeR R package:** defining rules from geno-pheno data
- 4. 2026 planning**
  - Manuscript plans
  - Meeting schedule
  - Events

# Agenda

- 1. Online resources:** where to find things
- 2. AMRrules Python package:** applying rules to interpret genotypes
- 3. AMRrulemakeR R package:** defining rules from geno-pheno data
- 4. 2026 planning**
  - Manuscript plans
  - Meeting schedule
  - Events

## ESGEM-AMR Working Group

This repository is the home of the ESGEM-AMR Working Group, which focuses on curating organism-specific rule sets for interpreting AMR genotypes under the [AMRrules](#) framework.



The goal of AMRrules is to develop interpretive standards for AMR genotypes, akin to the interpretive standards developed by [EUCAST](#) and [CLSI](#) for antimicrobial susceptibility phenotyping.

An overview of the concept, with the first release of AMRrules software and rules for ESKAPEE organisms, is available in the [AMRrules](#) repository.

We have partnered with [ESGEM](#), the ESCMID Study Group on Epidemiological Markers, to form an ESGEM-AMR Working Group to curate organism-specific rule sets.

- Slides from the introductory webinars held May 14/15 are available [here](#).
- A detailed description of the Working Group and the overall AMRrules approach is available [here](#), including scope, plans and timeline.
- Technical guidance for curation of rule sets is available [here](#), this is a work in progress and will be refined as we go.
- The rule specification template is [here \(v0.6, under active development\)](#).
- Validated rules are available in the [AMRrules](#) repository, together with a software package to apply these rules to interpret AMRfinderplus output.

## Membership

The convenors of the ESGEM-AMR Working Group are Kat Holt (LSHTM), Natacha Couto (ESGEM Chair), and Jane Hawkey (Monash, leading bioinformatics development).

A call for members was launched at the ESGEM General Meeting on April 29, 2024 and closed June 2. Over 120 applications were received and most of these have been invited to join organism-focused subgroups. We have now grown to 175 members (listed [below](#)).

Additional requests to join ESGEM-AMR will be considered periodically. In the meantime you may register your interest and let us know what organism/s you have expertise in, using [this form](#).

## Member Resources

- AMRrules Spec - v0.6 [\[google sheet\]](#)
- AMRrules Technical Guidance - v1.6 [\[PDF\]](#)
- AMRgen R package for analysing matched genotype/phenotype data [\[AMRgen repo\]](#)
- Code and tools for accessing public AMRfinderplus + AST data [\[datacuration repo\]](#)
- CARD/Antimicrobial Resistance Ontology (ARO): [\[browser\]](#)
- Variant specification: [HGVS](#)
- AMR rules syntax: [\[this repo\]](#)
- EUCAST: [\[Breakpoints\]](#) [\[Expected Resistance\]](#)
- NCBI AMRfinderplus: [\[software\]](#)
- NCBI refgene (AMR Gene Catalog): [\[browser\]](#)
- NCBI AMR Reference Gene Hierarchy: [\[browser\]](#) [\[TXT file download\]](#)
- NCBI refseq: [\[browser\]](#)
- NCBI AST data: [\[browser\]](#)
- NCBI AST data submission: [\[info\]](#) [\[submission format\]](#)
- NCBI Taxonomy: [\[browser\]](#)
- Introductory webinar slides: [\[PDF\]](#)
- Kickoff meeting slides: [\[PDF\]](#)

## Automatic validation of rules

Before submitting rules for review, they should be first checked using the Python [rulevalidator](#).

## Member List (by subgroup)

**Data & Tools** - Jane Hawkey/Kat Holt, Andrew McArthur, Finlay Maguire, Michael Feldgarden, Brody Duncan, Kristy Horan, Leonid Chindelevitch, Kara Tsang, Amogelang Raphenya, Dag Harmsen, Emily Bordeleau, Mackenzie Wilke, Romain Pogorelcnik, Yu Wan, Zoe Dyson, Bogdan Iorga, Derek Sarovich, John Rossen, Silvia Argimon, Charlene Rodrigues, Rolf Kaas, Nick Duggett, Louise Teixeira Cerdeira, Matthijs Berends, Adrian Egli, João Perdigão, Tiffany Ta, Karyn Mukiri, Chiara Crestani

**Enterococcus** - Francesc Coll, Thomas Demuyser, Ana R. Freitas, Guido Werner, Precious Osadebamwen, Theo Gouliouris, Fiona Walsh, Valeria Bortolaia, Basil Britto Xavier, Helena Seth-Smith

**Staphylococcus** - Natacha Couto, Birgitta Duim, Valeria Bortolaia, Sarah Baines, Sandra Reuter, Assaf Rokney, Holly Grace Espiriu, Manal AbuOun, Sankarganesh Jeyaraj, Robert Kozak, Nick Duggett, Birgit Strommenger, Lina Cavaco, Varun Shamanna, Sabrina Di Gregorio, Teresa Ribeiro, Tim Read, Georgina Lewis-Woodhouse

# AMRrules Documentation

<https://amrrules.readthedocs.io/>

(May 2025 beta release)  
(core genes only, no genome report)

The screenshot shows a web-based documentation page for the AMRrules Python package. At the top, there's a navigation bar with a menu icon, a download icon, a search icon, and a user profile icon. The main content area has a header 'AMRrules' with a subtitle 'AMRrules encode organism-specific rules for the interpretation of AMR genotype data, and are curated by organism experts belonging to ESGEM-AMR, a working group of ESGEM, the ESCMID Study Group on Epidemiological Markers. The rule specification is available in this Google sheet (v0.6, guidance on tab 2).'. Below this, there's a section about the standardization of antimicrobial susceptibility testing (AST) rules and how they apply to WGS data. Another section discusses the Python package's compatibility with NCBI resources like AMRFinderplus and ResFinder. The bottom section is about the initial rule curation for core genes and expected resistances.

## AMRrules

AMRrules encode organism-specific rules for the interpretation of AMR genotype data, and are curated by organism experts belonging to [ESGEM-AMR](#), a working group of [ESGEM](#), the ESCMID Study Group on [Epidemiological Markers](#). The rule specification is available in [this Google sheet](#) (v0.6, guidance on tab 2).

Organism-specific interpretation of antimicrobial susceptibility testing (AST) data is standard in clinical microbiology, with rules regularly reviewed by expert committees of [EUCAST](#) and [CLSI](#). We aim to provide an analogous resource to support organism-specific interpretation of antimicrobial resistance (AMR) genotypes derived from pathogen whole genome sequence (WGS) data.

This AMRrules Python package includes the rules themselves (see `rules/` directory) as well as code to apply the rules to interpret AMR genotypes (currently limited to [AMRFinderplus](#) output), generating informative genome reports that capture expert knowledge about how core and acquired genes and mutations contribute to antimicrobial susceptibility.

We are focusing early development on compatibility with NCBI resources (i.e. the [AMRFinderplus](#) genotyping tool, and the associated NCBI databases including [AMR refgene](#), [AMR Reference Gene Hierarchy](#), and the [Reference HMM Catalog](#)). In future we plan for interoperability with [CARD](#) and [ResFinder](#) (and other tools based on these), using [hAMRonization](#).

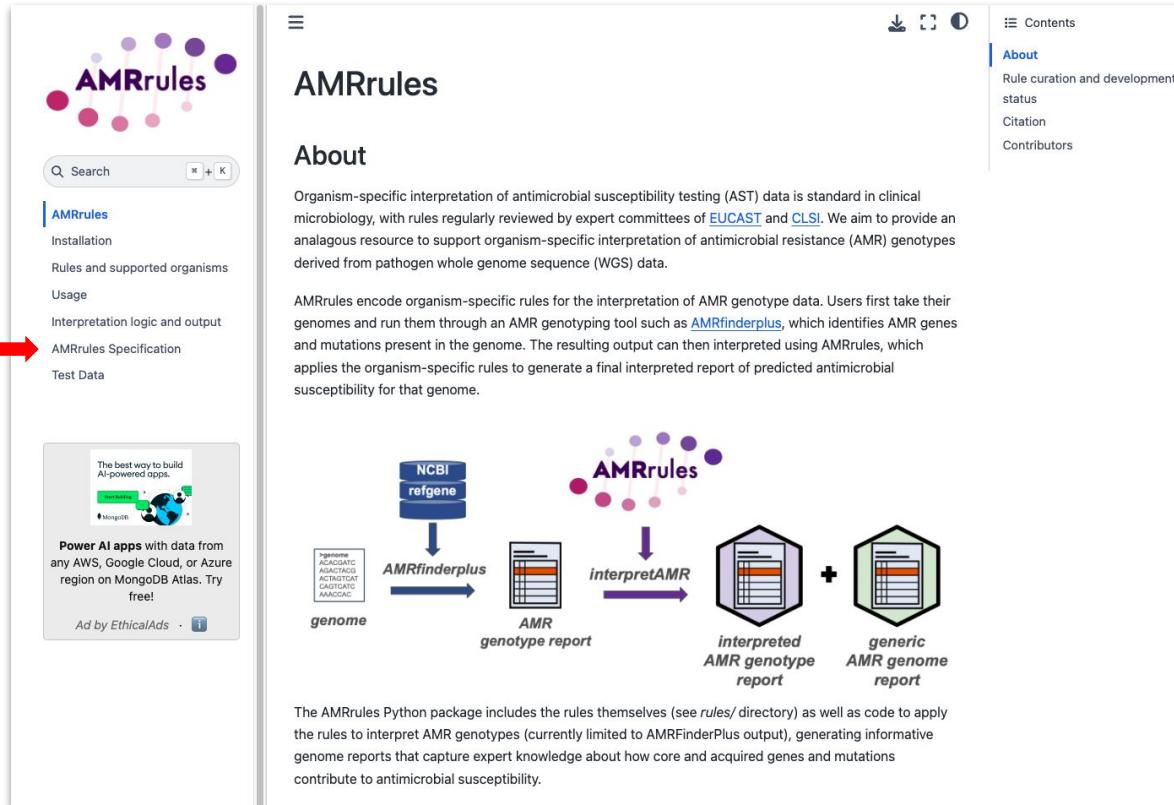
Initial rule curation has focused on defining rules for the interpretation of core genes and expected resistances, but acquired genes and mutations are included for some organisms already and will be added to others as the necessary data to define them accurately is accumulated and curated by the ESGEM-AMR working group.

# AMRrules Documentation

(current development branch)  
(use this one!)

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/](https://amrrules.readthedocs.io/en/genome_summary_report_dev/)

Rule specification



# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/specification.html](https://amrrules.readthedocs.io/en/genome_summary_report_dev/specification.html)

The screenshot shows the 'Rule Specification' page of the AMRrules documentation. The left sidebar includes links for AMRrules, Installation, Rules and supported organisms, Usage, Interpretation logic and output, AMRrules Specification (which is currently selected), and Test Data. A sidebar advertisement for APM insights is present. The main content area features a 'Rule Specification' section with a detailed description of how interpretive rules should be encoded in the AMRrules format. It also includes sections for the 'AMRrules template (Google sheet)' and a 'Full list of fields'. A table provides a detailed view of the required fields, their status, descriptions, reference standards, and links. A red arrow points from the text 'definitions' to the 'AMRrules template (Google sheet)' link in the sidebar. Another red arrow points from the text 'variant syntax' to the 'Full list of fields' section.

Rule Specification

This section details how interpretive rules should be encoded in the AMRrules format. The syntax for specifying different types of variants to which a rule should be applied is given in the next section.

On this page you will find the full list of fields (indicating which external databases or ontologies apply to each field, along with a description and guidance on defining/interpreting each field), as well as bespoke AMRrules-specific controlled vocabulary for some fields.

### AMRrules template (Google sheet)

The rule specification is also available in a [Google sheet that includes the AMRrules template](#), with allowed values encoded in drop-down menus, to facilitate rule curation.

### Full list of fields

The full list of fields is below, with guidelines on how each field should be specified and interpreted.

[Download](#)

Required fields	status	description	reference standard	reference link
ruleID	required	unique identifier for this rule {values listed in 'organism subgroup codes'}	AMRrules	'organism subgroup codes' tab

Contents

- Rule Specification
  - AMRrules template (Google sheet)
- Full list of fields
- Controlled vocabularies
  - Variation type
  - Evidence codes
  - Evidence grade
  - Evidence limitations
  - Breakpoint condition
  - Organism code
- Variant Specification
  - Syntax for mutations
    - AMRrules-specific syntax
  - Explanation of 'mutation' syntax relevant to known AMR variants
  - Combinatorial rules

# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/specification.html#controlled-vocabularies](https://amrrules.readthedocs.io/en/genome_summary_report_dev/specification.html#controlled-vocabularies)

The screenshot shows a web page with a sidebar on the right. The main content area is titled 'Controlled vocabularies' and contains a section titled 'Variation type'. Below this, there is a table with six rows, each listing a value allowed in the 'variation type' column and the conditions under which the specified AMRrule applies. The sidebar on the right is titled 'Contents' and lists various sections, with 'Variation type' being the currently selected one, indicated by a blue border.

Values allowed in variation type column	The specified AMRrule applies if...	Notes or source
Gene presence detected	...the gene specified in the 'gene' column is detected as being present.	hAMRonization
Protein variant detected	...the protein variant specified in the 'mutation' column is detected in the specified 'gene'.	hAMRonization
Nucleotide variant detected	...the nucleotide variant specified in the 'mutation' column is detected in the specified 'gene'.	hAMRonization
Promoter variant detected	...the promoter variant specified in the 'mutation' column is detected in the specified 'gene'.	NCIT:C190205
Inactivating mutation detected	...the gene specified in the 'gene' column is inactivated by any type of mechanism (e.g. frameshift, internal stop, deletion, truncation), in the amino acid range specified in the 'mutation' column (or anywhere in the gene, if the 'mutation' column is blank i.e. '-').	NCIT:C178119
Gene copy number variant detected	...the gene specified in the 'gene' column is detected in at least the minimum number of copies specified in the 'mutation' column.	NCIT:C189957

← definitions:  
variation type

# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/specification.html#evidence-codes](https://amrrules.readthedocs.io/en/genome_summary_report_dev/specification.html#evidence-codes)

## Evidence codes

Specified using the [Evidence and Conclusion Ontology \(ECO\)](#), this field indicates the nature of the evidence supporting the rule. More than one can be listed, and the field should include all forms of evidence available to support the rule (multiple entries separated with ', ').

Any ECO codes can be used, but curators are encouraged to choose from the subset listed here, which covers the types of evidence typically available to support resistance mechanisms in bacteria. Note the literature source for each type of evidence noted here should be indicated in the [PMID](#) field.

ECO:0001091	knockout phenotypic evidence	ECO:0001091 knockout phenotypic evidence	E.g. evidence that knocking out the proposed AMR gene in a phenotypically resistant strain results in loss of resistance
ECO:0000012	functional complementation evidence	ECO:0000012 functional complementation evidence	E.g. evidence that, when a gene knockout results in change from R to S, the phenotype is reversed (resistance is restored) when the gene is reintroduced
ECO:0001113	point mutation phenotypic evidence	ECO:0001113 point mutation phenotypic evidence	E.g. for a mutation, evidence that this specific mutation is associated with a change in susceptibility phenotype
ECO:0000024	protein-binding evidence	ECO:0000024 protein-binding evidence	E.g. evidence that the gene product binds to this drug
ECO:0001034	crystallography evidence	ECO:0001034 crystallography evidence	E.g. structural evidence from crystallography that the mutated position in this gene product interacts with the drug

### Contents

- Rule Specification
- AMRrules template (Google sheet)
- Full list of fields
- Controlled vocabularies**
- Variation type
- Evidence codes**
- Evidence grade
- Evidence limitations
- Breakpoint condition
- Organism code
- Variant Specification
- Syntax for mutations
  - AMRrules-specific syntax
  - Explanation of 'mutation' syntax relevant to known AMR variants
- Combinatorial rules

definitions:  
evidence code

# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/specification.html#evidence-grade](https://amrrules.readthedocs.io/en/genome_summary_report_dev/specification.html#evidence-grade)

The screenshot shows a documentation page for AMRrules. On the left, the main content area is titled "Evidence grade". It contains several paragraphs of text explaining the concept of evidence grade, its purpose, and its relationship to the GRADE framework. Below this is a table comparing four evidence grades: high, moderate, low, and very low. The "Evidence grade" column lists the grade names, the "What it means" column provides a detailed description of each grade's meaning, and the "Use this when" column provides guidance on when to use each grade. On the right, a sidebar titled "Contents" lists various sections of the documentation, with "Evidence grade" highlighted in blue. A red arrow points from the sidebar back to the "Evidence grade" section in the main content area.

Evidence grade

This field indicates the expert curators' overall assessment of the level of support provided by all evidence considered. It is modelled on the [GRADE](#) (Grading of Recommendations, Assessment, Development, and Evaluation) approach to assessing the certainty of evidence to guide decision making in healthcare.

AMRrules aims to provide rules to interpret all markers that have been detected in a given species, but in many cases the evidence can be quite limited. The [evidence grade](#) field gives users an overall guide to the strength of evidence, and the [evidence limitations](#) field highlights what kind of evidence is lacking.

Note that if no experimental evidence is available, the rule should NOT be graded as 'high', even if there is strong evidence of statistical association between genotype and phenotype in natural populations. (Future updates to the rule specification will include additional fields to record quantitative details of genotype/phenotype associations.)

There are four possible 'grades' in AMRrules, these are listed below with guidance on what they mean in the context of AMRrules (modelled on the [GRADE](#) framework).

Evidence grade	What it means	Use this when
high	The curators are confident in the categorisation, and believe that the likelihood that the effect will be substantially different from this is low.	Experimental evidence provides strong support for the interpretation of this gene/variant in this species for this drug. If there is statistical geno/pheno evidence available, it supports this interpretation.
moderate	The curators believe that the categorisation most likely reflects the true effect, and the likelihood that the effect will be substantially different is moderate.	There is good evidence to support the interpretation of this gene/variant in this species for this drug, but there is some uncertainty (e.g. lack of direct evidence in this organism although evidence from related organisms is convincing; or there is good statistical geno/pheno evidence but no experimental evidence of mechanism).
low	The curators believe that the categorisation is likely to be wrong, and the likelihood that the effect will be substantially different is high.	There is no evidence to support the interpretation of this gene/variant in this species for this drug, and the evidence available is not convincing enough to rule out other interpretations.
very low	The curators believe that the categorisation is almost certainly wrong, and the likelihood that the effect will be substantially different is very high.	There is no evidence to support the interpretation of this gene/variant in this species for this drug, and the evidence available is not convincing enough to rule out other interpretations.

Contents

- Rule Specification
- AMRrules template (Google sheet)
- Full list of fields
- Controlled vocabularies**
- Variation type
- Evidence codes
- Evidence grade**
- Evidence limitations
- Breakpoint condition
- Organism code
- Variant Specification
- Syntax for mutations
- AMRrules-specific syntax
- Explanation of 'mutation' syntax relevant to known AMR variants
- Combinatorial rules

← definitions:  
evidence grade

# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/specification.html](https://amrrules.readthedocs.io/en/genome_summary_report_dev/specification.html)

The screenshot shows the 'Rule Specification' page of the AMRrules documentation. The left sidebar includes links for AMRrules, Installation, Rules and supported organisms, Usage, Interpretation logic and output, AMRrules Specification (which is active), and Test Data. A sidebar advertisement for APM insights is present. The main content area features a heading 'Rule Specification' and a paragraph explaining how interpretive rules should be encoded in the AMRrules format. It also mentions the 'AMRrules template (Google sheet)' and provides a 'Full list of fields'. A red arrow points from the text 'rule template' to the 'AMRrules template (Google sheet)' link in the sidebar table of contents.

rule template

Required fields	status	description	reference standard	reference link
ruleID	required	unique identifier for this rule {values listed in 'organism subgroup codes'}	AMRrules	'organism subgroup codes' tab

# AMRrules template

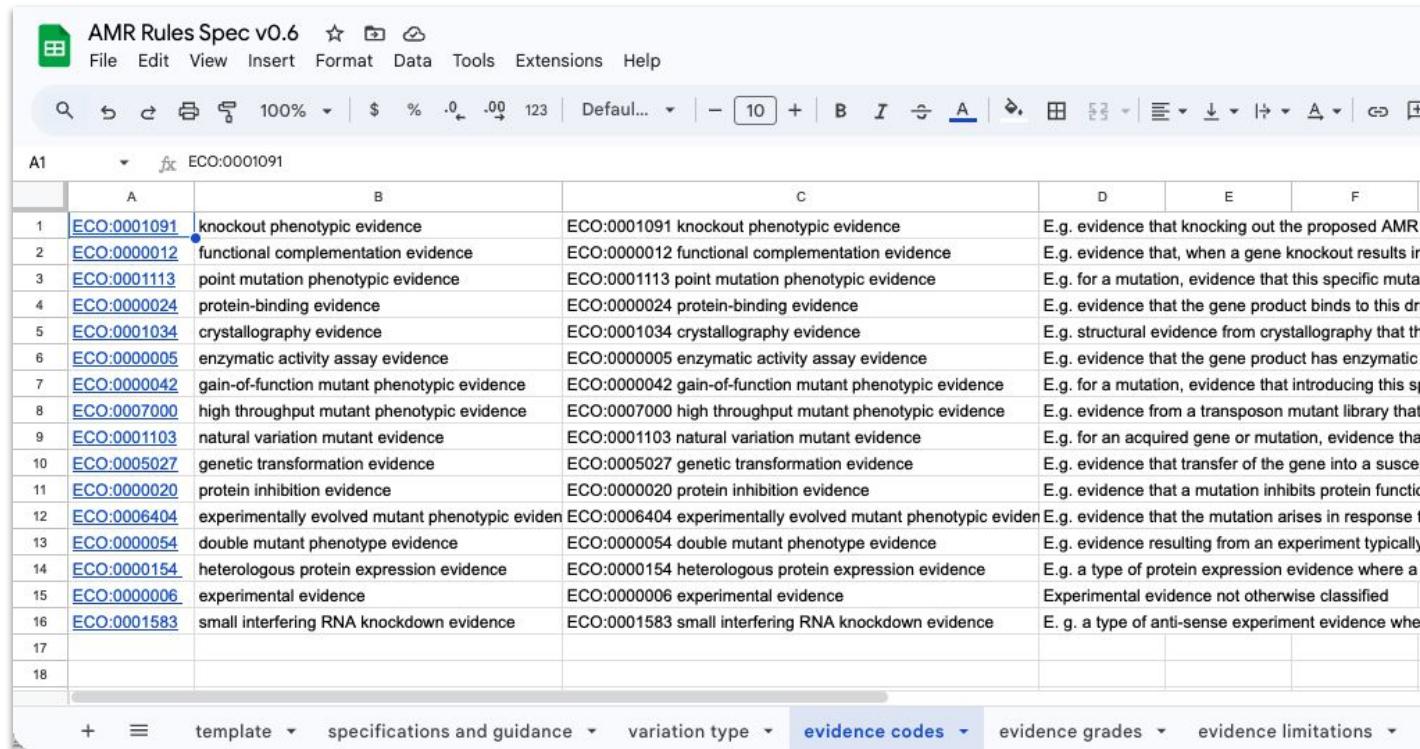
[https://bit.ly/AMRrules\\_spec06](https://bit.ly/AMRrules_spec06)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ruleID	txid	organism	gene	nodeID	protein accession	HMM accession	nucleotide accession	ARO accession	mutation	variation type	gene context	drug	drug class
2	required	required	required	required	uniquely identify the gene using AT LEAST ONE NCBI accession: nodeID (preferred) or protein sequence accession (refseq preferred, or GenBank) or HMM accession, or nucleotide sequence accession with coordinates (for nucleotide variants e.g. 23S or promoter regions)					optional (recommended)	required	required	required	require ONE OF: drug or drug class
3	KPN0001	573	s_Klebsiella pneumoniae	blaSHV	blaSHV	NF000285.3	-		ARO:3000015	-	Gene presence detected	core	-	penicillin beta-lactamase
4	KPN0002	573	s_Klebsiella pneumoniae	oqxA	oqxA	NF000272.1	-		ARO:3003922	-	Gene presence detected			fluoroquinolone antibiotic
5	KPN0003	573	s_Klebsiella pneumoniae	oqxB	oqxB	NF000037.1	-		ARO:3003923	-	Protein variant detected			fluoroquinolone antibiotic
6	KPN0004	573	s_Klebsiella pneumoniae	fosA5_fam	fosA5_fam	NF040540.1	-		-	-	Promoter variant detected		isfomycin	-
7	KPN0005	573	s_Klebsiella pneumoniae	fosA5	fosA5	WP_012579083.1	-		ARO:3003209	-	Inactivating mutation detected			isfomycin
8	KPN0006	573	s_Klebsiella pneumoniae	fosA6	fosA6	WP_069174570.1	-		ARO:3004111	-	Gene copy number variant detected			isfomycin
9	KPN0007	573	s_Klebsiella pneumoniae	fosA10	fosA10	WP_004214174.1	-		-	-	Nucleotide variant detected in multi-copy gene			isfomycin
10	KPN0008	573	s_Klebsiella pneumoniae	oqxA	oqxA	NF000272.1	-		ARO:3003922	-	Low frequency variant detected			isfomycin
11	KPN0009	573	s_Klebsiella pneumoniae	oqxB	oqxB	NF000037.1	-		ARO:3003923	-	Combination			isfomycin
12	KPN0010	573	s_Klebsiella pneumoniae	gyrA	gyrA	WP_117036963.1	-		-	p.Ser83Tyr	Protein variant detected			ciprofloxacin
13	KPN0011	573	s_Klebsiella pneumoniae	qnrA1	qnrA1	WP_012579084.1	-		-	-	Gene presence detected	acquired		ciprofloxacin

# AMRrules template

[https://bit.ly/AMRrules\\_spec06](https://bit.ly/AMRrules_spec06)

*controlled vocabulary definitions also available as tabs in this document*



The screenshot shows a Google Sheets spreadsheet with the title "AMR Rules Spec v0.6". The table has columns A through F. Column A contains ECO codes, column B contains evidence types, column C contains detailed descriptions, and column D contains examples. The rows list various types of evidence, such as knockout phenotypic evidence, functional complementation evidence, and protein-binding evidence. The "evidence codes" tab is highlighted with a red arrow at the bottom.

A	B	C	D	E	F
1 ECO:0001091	knockout phenotypic evidence	ECO:0001091 knockout phenotypic evidence	E.g. evidence that knocking out the proposed AMR		
2 ECO:0000012	functional complementation evidence	ECO:0000012 functional complementation evidence	E.g. evidence that, when a gene knockout results in		
3 ECO:0001113	point mutation phenotypic evidence	ECO:0001113 point mutation phenotypic evidence	E.g. for a mutation, evidence that this specific mutati		
4 ECO:0000024	protein-binding evidence	ECO:0000024 protein-binding evidence	E.g. evidence that the gene product binds to this dru		
5 ECO:0001034	crystallography evidence	ECO:0001034 crystallography evidence	E.g. structural evidence from crystallography that the		
6 ECO:0000005	enzymatic activity assay evidence	ECO:0000005 enzymatic activity assay evidence	E.g. evidence that the gene product has enzymatic a		
7 ECO:0000042	gain-of-function mutant phenotypic evidence	ECO:0000042 gain-of-function mutant phenotypic evidence	E.g. for a mutation, evidence that introducing this sp		
8 ECO:0007000	high throughput mutant phenotypic evidence	ECO:0007000 high throughput mutant phenotypic evidence	E.g. evidence from a transposon mutant library that		
9 ECO:0001103	natural variation mutant evidence	ECO:0001103 natural variation mutant evidence	E.g. for an acquired gene or mutation, evidence that		
10 ECO:0005027	genetic transformation evidence	ECO:0005027 genetic transformation evidence	E.g. evidence that transfer of the gene into a suscep		
11 ECO:0000020	protein inhibition evidence	ECO:0000020 protein inhibition evidence	E.g. evidence that a mutation inhibits protein functio		
12 ECO:0006404	experimentally evolved mutant phenotypic eviden	ECO:0006404 experimentally evolved mutant phenotypic eviden	E.g. evidence that the mutation arises in response to		
13 ECO:0000054	double mutant phenotype evidence	ECO:0000054 double mutant phenotype evidence	E.g. evidence resulting from an experiment typically		
14 ECO:0000154	heterologous protein expression evidence	ECO:0000154 heterologous protein expression evidence	E.g. a type of protein expression evidence where a g		
15 ECO:0000006	experimental evidence	ECO:0000006 experimental evidence	Experimental evidence not otherwise classified		
16 ECO:0001583	small interfering RNA knockdown evidence	ECO:0001583 small interfering RNA knockdown evidence	E. g. a type of anti-sense experiment evidence wher		
17					
18					

# AMRrulevalidator

<https://github.com/AMRverse/AMRrulevalidator>

## AMRrulevalidator

The AMRrulevalidator package provides tools for validating [AMRrules files](#) according to the current specification ([v0.6](#)).

As part of the install, AMRrulevalidator will download the current CARD ontology and AMRFinderPlus resources to validate against.

The `validate` subcommand will print to stdout a summary of checks that have been completed, and whether they've passed or failed, and will write out a version of the rules file where cells are flagged with values that need to be checked.

The `clean` subcommand will write out a cleaned version of a rules file after all values have been checked, which will be ready for integration into the AMRrules interpretation engine.

## Installation

AMRrulevalidator is compatible with Python >= 3.8. The only dependency is [obonet v1.1.1](#).

The easiest installation method is via pip from GitHub, which will install all required dependencies for you.

## Installation with pip

```
# Optional: create a conda environment to install package into
conda create -n amrrulevalidator

conda activate amrrulevalidator

conda install pip

# Install with pip from GitHub
pip install git+https://github.com/AMRverse/AMRrulevalidator.git

# Download CARD and AMRFinderPlus ontology files
amrrule update-resources
```

## Usage

### Validating a rules file

To validate a draft AMRrules file:

```
amrrule validate --input path/to/draft_rules.tsv --output path/to/validated_rules.tsv
```

This will:

1. Check the input file against the current AMRrules specification
2. Generate a validated output file with annotations for any errors
3. Print a summary of validation results to the console

### Validation checks

During validation, the script annotates problematic values in the output file to help identify and fix issues:

- **ENTRY MISSING**: Indicates that a required value is missing in a field where a value is expected.
- **CHECK VALUE: [value]**: Indicates that the existing value doesn't match the expected format or isn't in the list of allowed values.

```
Checking ruleID column...
✓ All ruleIDs are valid (prefix: STA)

Checking txid column...
✗ 2 rows have failed the check
Txids must be present, not 'NA' or '-'. Txids should be in the NCBI taxonomy list, as per file resources/ncbi_taxonomy.tsv.
Row 9: 1281 is not a valid NCBI taxonomic ID
Row 14: 1281 is not a valid NCBI taxonomic ID
```

### Summary of checks:

```
✓ Passed: 9
- ruleID
- organism
- gene
- ARO accession
- gene context
- drug and drug class
- phenotype
- PMID
- evidence grade and limitations
✗ Failed: 10
- txid
- txid and organism
- gene accessions
- variation type
- variation type mutation concordance
- clinical category
- breakpoint
- breakpoint standard
- breakpoint condition
- evidence code
```

Checked against AMRFinderPlus database version: 2025-07-16.1

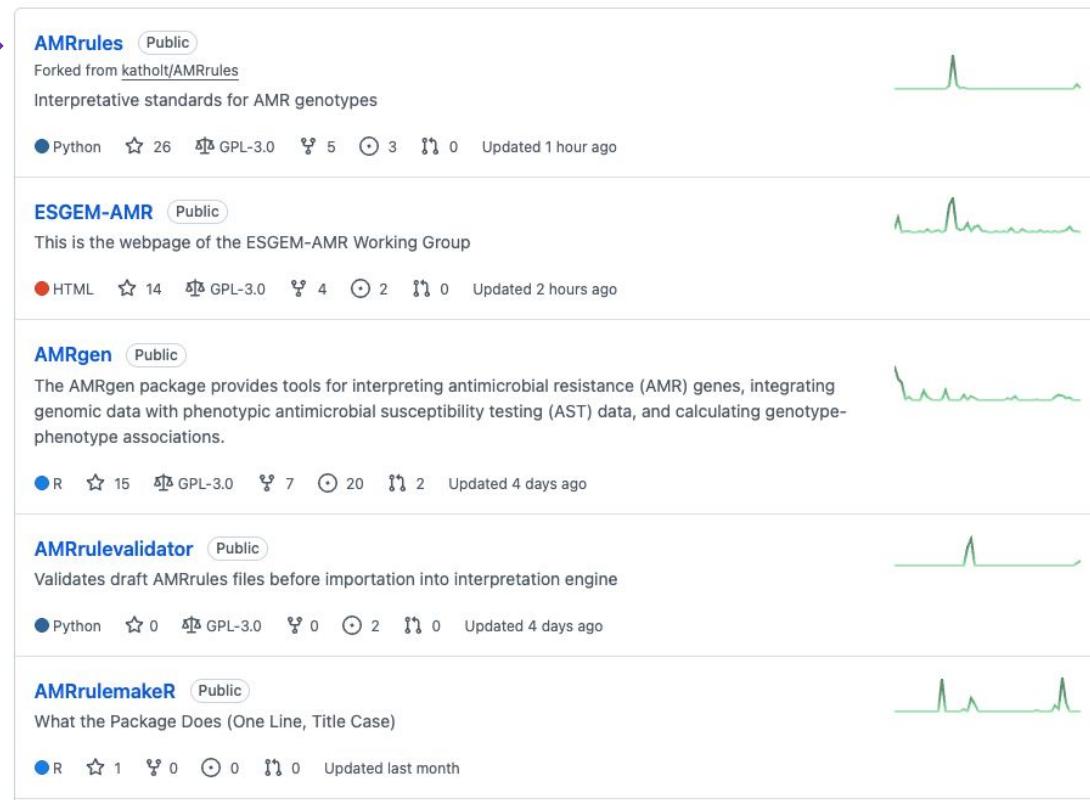
*Looks up values in  
CARD, NCBI and  
AMRFinderplus  
gene hierarchy*

*Full details of  
checks in README*



## Our home on github: <https://github.com/AMRverse/>

docs, rule spec  
(under development branch)



AMRrules Public  
Forked from katholt/AMRrules  
Interpretative standards for AMR genotypes

Python ⭐ 26 GPL-3.0 5 3 0 Updated 1 hour ago

ESGEM-AMR Public  
This is the webpage of the ESGEM-AMR Working Group

HTML ⭐ 14 GPL-3.0 4 2 0 Updated 2 hours ago

AMRgen Public  
The AMRgen package provides tools for interpreting antimicrobial resistance (AMR) genes, integrating genomic data with phenotypic antimicrobial susceptibility testing (AST) data, and calculating genotype-phenotype associations.

R ⭐ 15 GPL-3.0 7 20 2 Updated 4 days ago

AMRrulevalidator Public  
Validates draft AMRrules files before importation into interpretation engine

Python ⭐ 0 GPL-3.0 0 2 0 Updated 4 days ago

AMRrulemakeR Public  
What the Package Does (One Line, Title Case)

R ⭐ 1 0 0 0 Updated last month

# Agenda

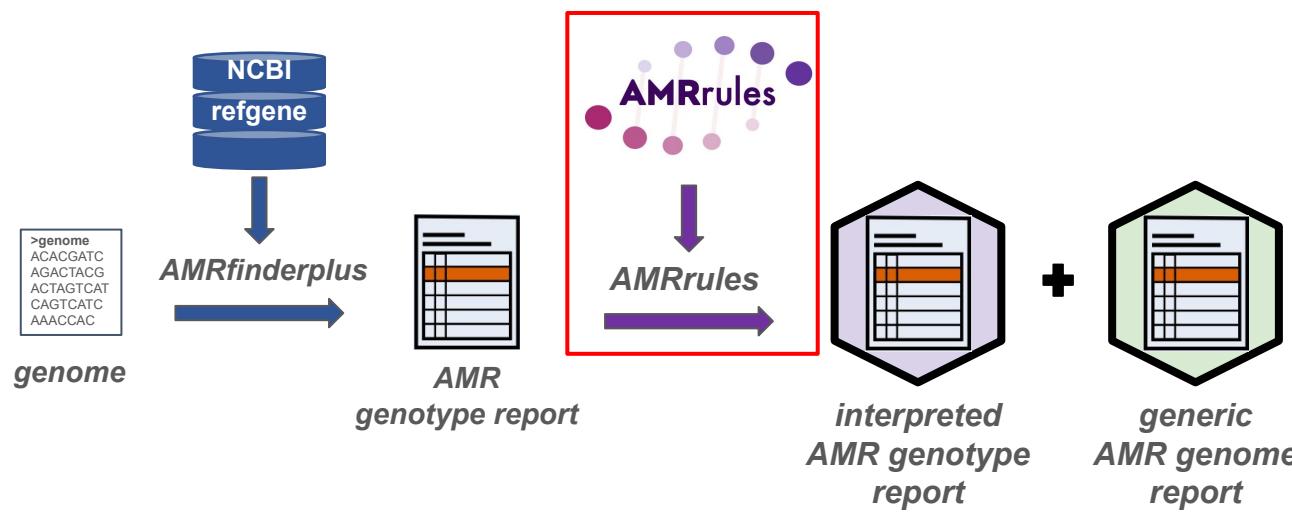
1. **Online resources:** where to find things
2. **AMRrules Python package:** applying rules to interpret genotypes
3. **AMRrulemakeR R package:** defining rules from geno-pheno data
4. **2026 planning**
  - Manuscript plans
  - Meeting schedule
  - Events

# AMRrules Python package

Development branch: [https://github.com/AMRverse/AMRrules/tree/genome\\_summary\\_report\\_dev](https://github.com/AMRverse/AMRrules/tree/genome_summary_report_dev)

## Package includes

- rules
- software to interpret AMRfinderplus genotype results using the rules



Rules in first beta release:

- [Acinetobacter baumannii](#)
- [Enterobacter](#)
- [Enterococcus faecalis](#)
- [Enterococcus faecium](#)
- [Escherichia coli](#)
- [Klebsiella pneumoniae](#)
- [Neisseria gonorrhoeae](#) (acquired resistances)
- [Pseudomonas aeruginosa](#)
- [Salmonella](#)
- [Staphylococcus aureus](#)
- [Yersinia](#)

## How to test

### Installation:

```
conda create -n amrrules_beta -c bioconda  
python=3.12 pip
```

```
conda activate amrrules_beta
```

```
git clone  
https://github.com/interpretAMR/AMRrules
```

```
cd AMRrules
```

### Switch to development branch:

```
git checkout genome_summary_report_dev
```

```
make dev
```

### Download resources (only required on initial install or when wanting to update NCBI/CARD dbs):

```
amrrules --download-resources
```

### Rules for v1 release

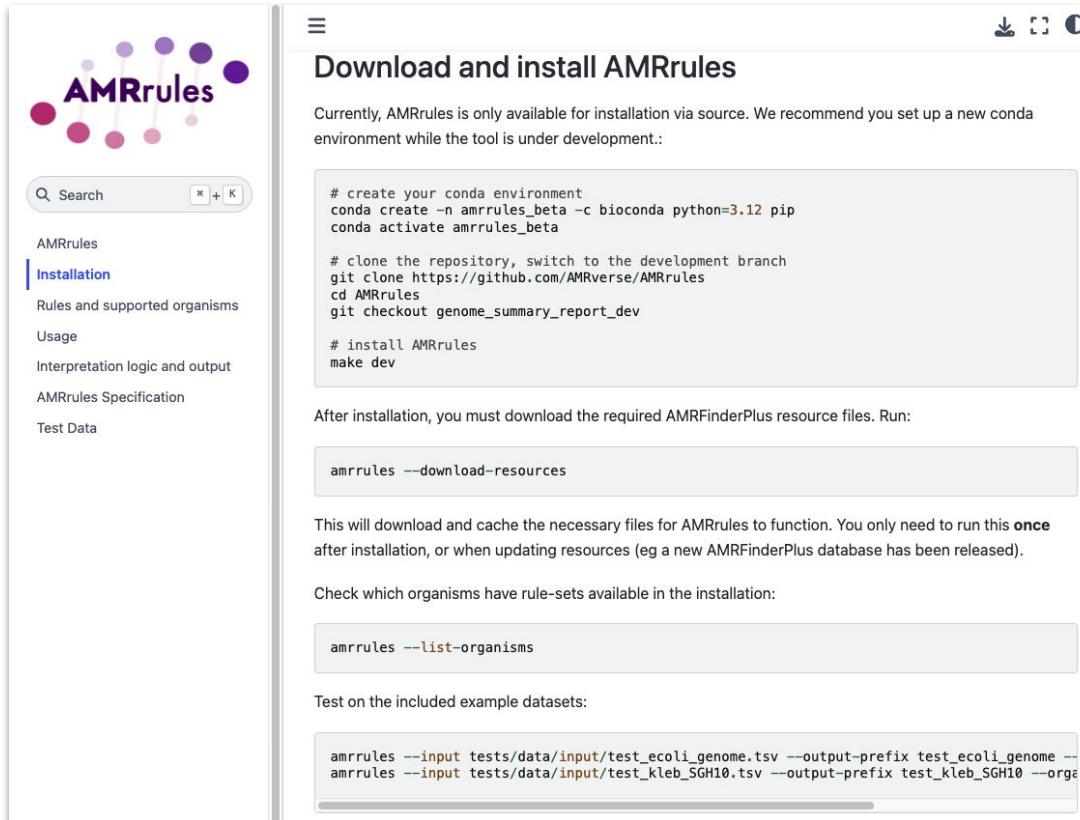
### (v0.6 spec)

- *Acinetobacter baumannii*
- *Enterobacter*
- *Enterococcus faecalis*
- *Enterococcus faecium*
- *Escherichia coli*
- *Klebsiella pneumoniae*
- *Pseudomonas aeruginosa*
- *Salmonella*
- *Staphylococcus aureus*
- *Yersinia*
- *Neisseria gonorrhoeae* (acquired)
- *Klebsiella oxytoca*
- *Bordetella*
- *Shewanella*
- *Burkholderia*
- *Legionella*
- *Achromobacter xylosoxidans*
- *Proteus mirabilis*
- *Neisseria meningitidis*

# AMRrules Documentation

[https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/installation.html](https://amrrules.readthedocs.io/en/genome_summary_report_dev/installation.html)

*How to test*



The screenshot shows a web-based documentation interface for AMRrules. On the left, there's a sidebar with a logo featuring purple circles connected by lines, a search bar, and a navigation menu with the following items: AMRrules, Installation (which is highlighted), Rules and supported organisms, Usage, Interpretation logic and output, AMRrules Specification, and Test Data.

**Download and install AMRrules**

Currently, AMRrules is only available for installation via source. We recommend you set up a new conda environment while the tool is under development.:

```
# create your conda environment
conda create -n amrrules_beta -c bioconda python=3.12 pip
conda activate amrrules_beta

# clone the repository, switch to the development branch
git clone https://github.com/AMRverse/AMRrules
cd AMRrules
git checkout genome_summary_report_dev

# install AMRrules
make dev
```

After installation, you must download the required AMRFinderPlus resource files. Run:

```
amrrules --download-resources
```

This will download and cache the necessary files for AMRrules to function. You only need to run this **once** after installation, or when updating resources (eg a new AMRFinderPlus database has been released).

Check which organisms have rule-sets available in the installation:

```
amrrules --list-organisms
```

Test on the included example datasets:

```
amrrules --input tests/data/input/test_ecoli_genome.tsv --output-prefix test_ecoli_genome --
amrrules --input tests/data/input/test_kleb_SGH10.tsv --output-prefix test_kleb_SGH10 --orga
```

# Interpretation engine: running

[bit.ly/AMRules](https://bit.ly/AMRules)

```
amrrules --input tests/data/debug/WHO_alpha_2024_amrf.tsv --output-prefix test_gono_WHO_alpha  
--output-dir tests/data/output --organism 's__Neisseria gonorrhoeae' --annot-opts full --no-rule-interpretation nwtr
```

16 hits matched a rule and 0 hits did not match a rule.

Output written to tests/data/output/test\_gono\_WHO\_alpha\_interpreted.tsv.

Summary output written to tests/data/output/test\_gono\_WHO\_alpha\_genome\_summary.tsv.

```
amrrules --input tests/data/input/test_data_amrfp_multSpp.tsv --output-prefix test_multSpp  
--output-dir tests/data/output --organism-file tests/data/input/test_data_sppCalls.tsv
```

UserWarning: The following sample IDs from the organism file are not present in the input file:

NOT\_IN\_INPUT\_FILE

As there are no entries in the input file for these samples, they won't have interpretation results.

Please check your input file if this is not what you expect.

```
warnings.warn(f"The following sample IDs from the organism file are not present in the input file:\n{'\n'.join(missing_from_input)}\nsamples, they won't have interpretation results. Please check your input file if this is not what you expect.")
```

71 hits matched a rule and 89 hits did not match a rule.

Output written to tests/data/output/test\_multSpp\_interpreted.tsv.

Summary output written to tests/data/output/test\_multSpp\_genome\_summary.tsv.

## Organism file example

SAMD00028844	s__Acinetobacter baumannii
SAMD00000649	s__Escherichia coli
SAMD00055728	s__Klebsiella pneumoniae
SAMD00002817	s__Staphylococcus aureus
SAMD00019033	s__Pseudomonas aeruginosa
SAMEA1572980	s__Neisseria gonorrhoeae
SAMD00499917	s__Yersinia enterocolitica

↑  
Matches  
“Name” in  
AMRFP input

↑  
Valid organism  
from  
--organism flag

# Annotated AMRfinderplus report

[bit.ly/AMRrules](http://bit.ly/AMRrules)  
**(v1-in dev)**



AMRfinderplus

AMRrules

genotype interpretation

Gene symbol	Subclass	Hierarchy node	Variation type	mutation	ruleID	gene context	drug	drug class	phenotype	clinical category	evidence grade	version	organism
emrD	EFFLUX	emrD	Gene presence detected	-	-	-	-	-	-	-	-	0.1.0	-
oqxB19	PHENICOL/ QUINOLONE	oqxB19	Gene presence detected	-	KPN0003	core	ciprofloxacin	-	wildtype	S	moderate	0.1.0	s_Klebsiella pneumoniae
oqxA	PHENICOL/ QUINOLONE	oqxA	Gene presence detected	-	KPN0002	core	ciprofloxacin	-	wildtype	S	moderate	0.1.0	s_Klebsiella pneumoniae
blaSHV-11	BETA-LACTAM	blaSHV-11	Gene presence detected	-	KPN0001	core	-	penicillin beta-lactam	wildtype	R	high	0.1.0	s_Klebsiella pneumoniae
fosA	FOSFOMYCIN	fosA5_fam	Gene presence detected	-	KPN0004	core	fosfomycin	-	wildtype	S	moderate	0.1.0	s_Klebsiella pneumoniae

- Default: ‘minimal’ annotation (as shown here)
- ‘full’ option additionally contains:
  - breakpoint, breakpoint standard, evidence code, evidence limitations, PMID, rule curation note

# Mapping of AMRfinderplus results to AMRrules variation type

Variation type	AMRFP Method	Additional logic
Gene presence detected	EXACT, ALLELE, BLAST	
Protein variant detected	POINTX, POINTP	
Nucleotide variant detected	POINTN	
Promoter variant detected	POINTN	 symbol present before the nucleotide position
Inactivating mutation detected	INTERNAL_STOP, PARTIAL, POINTX	If POINTX, Subtype must be POINT_DISRUPT

# Mapping of AMRfinderplus results to AMRrules variation type

Gene symbol	Method	Hierarchy node	variation type	gene	mutation	ruleID
aac(3)-la	BLASTX	aac(3)-la	Gene presence detected	aac(3)-la	-	-
aadA1	EXACTX	aadA1	Gene presence detected	aadA1	-	-
sul1	EXACTX	sul1	Gene presence detected	sul1	-	-
parC_S84L	POINTX	parC	Protein variant detected	parC	p.Ser84Leu	-
adeC	PARTIAL_CONTIG_ENDX	adeC	Inactivating mutation detected	adeC	-	-
aph(6)-Id	EXACTX	aph(6)-Id	Gene presence detected	aph(6)-Id	-	-
aph(3")-lb	EXACTX	aph(3")-lb	Gene presence detected	aph(3")-lb	-	-
gyrA_S81L	POINTX	gyrA	Protein variant detected	gyrA	p.Ser81Leu	ACI0162
aph(3')-la	EXACTX	aph(3')-la	Gene presence detected	aph(3')-la	-	-
tet(B)	BLASTX	tet(B)	Gene presence detected	tet(B)	-	-
sul2	EXACTX	sul2	Gene presence detected	sul2	-	-
amvA	PARTIAL_CONTIG_ENDX	amvA	Inactivating mutation detected	amvA	-	-
blaADC	PARTIALX	blaADC	Inactivating mutation detected	blaADC	-	-
blaOXA	BLASTX	blaOXA-51_fam	Gene presence detected	blaOXA	-	ACI0146
ant(3")-lla	PARTIALX	ant(3")-lla	Inactivating mutation detected	ant(3")-lla	-	-

## Mapping of AMRfinderplus results to AMRrules variation type

Gene symbol	Method	Hierarchy node	variation type	gene	mutation	ruleID
aac(3)-la	BLASTX	aac(3)-la	Gene presence detected	aac(3)-la	-	-
aadA1	EXACTX	aadA1	Gene presence detected	aadA1	-	-
sul1	EXACTX	sul1	Gene presence detected	sul1	-	-
parC_S84L	POINTX	parC	Protein variant detected	parC	p.Ser84Leu	-
adeC	PARTIAL_CONTIG_ENDX	adeC	Inactivating mutation detected	adeC	-	-
aph(6)-Id	EXACTX	aph(6)-Id	Gene presence detected	aph(6)-Id	-	-
aph(3")-lb	EXACTX	aph(3")-lb	Gene presence detected	aph(3")-lb	-	-
gyrA_S81L	POINTX	gyrA	Protein variant detected	gyrA	p.Ser81Leu	ACI0162
aph(3')-la	EXACTX	aph(3')-la	Gene presence detected	aph(3')-la	-	-
tet(B)	BLASTX	tet(B)	Gene presence detected	tet(B)	-	-
sul2	EXACTX	sul2	Gene presence detected	sul2	-	-
amvA	PARTIAL_CONTIG_ENDX	amvA	Inactivating mutation detected	amvA	-	-
blaADC	PARTIALX	blaADC	Inactivating mutation detected	blaADC	-	-
blaOXA	BLASTX	blaOXA-51_fam	Gene presence detected	blaOXA	-	ACI0146
ant(3")-lla	PARTIALX	ant(3")-lla	Inactivating mutation detected	ant(3")-lla	-	-

## Mapping of AMRfinderplus results to AMRrules variation type

Gene symbol	Method	Hierarchy node	variation type	gene	mutation	ruleID
aac(3)-la	BLASTX	aac(3)-la	Gene presence detected	aac(3)-la	-	-
aadA1	EXACTX	aadA1	Gene presence detected	aadA1	-	-
sul1	EXACTX	sul1	Gene presence detected	sul1	-	-
parC_S84L	POINTX	parC	Protein variant detected	parC	p.Ser84Leu	-
adeC	PARTIAL_CONTIG_ENDX	adeC	Inactivating mutation detected	adeC	-	-
aph(6)-Id	EXACTX	aph(6)-Id	Gene presence detected	aph(6)-Id	-	-
aph(3")-lb	EXACTX	aph(3")-lb	Gene presence detected	aph(3")-lb	-	-
gyrA_S81L	POINTX	gyrA	Protein variant detected	gyrA	p.Ser81Leu	ACI0162
aph(3')-la	EXACTX	aph(3')-la	Gene presence detected	aph(3')-la	-	-
tet(B)	BLASTX	tet(B)	Gene presence detected	tet(B)	-	-
sul2	EXACTX	sul2	Gene presence detected	sul2	-	-
amvA	PARTIAL_CONTIG_ENDX	amvA	Inactivating mutation detected	amvA	-	-
blaADC	PARTIALX	blaADC	Inactivating mutation detected	blaADC	-	-
blaOXA	BLASTX	blaOXA-51_fam	Gene presence detected	blaOXA	-	ACI0146
ant(3")-lla	PARTIALX	ant(3")-lla	Inactivating mutation detected	ant(3")-lla	-	-

# Example genome summary report

*Klebsiella pneumoniae* SGH10 (ampR)

drug	drug class	clinical category	phenotype
ciprofloxacin	fluoroquinolone antibiotic	S	wildtype
-	penicillin beta-lactam	R	wildtype
fosfomycin	phosphonic acid antibiotic	S	wildtype



**Class field = CARD term**

- If there is a rule defined for a class, this field is taken from the ‘drug class’ field in the rule
- If there is rule defined for a drug, the ‘drug’ specified in the rule is mapped to a CARD class
- If there is no rule, the ‘Subclass’ field in AMRfinderplus is mapped to a CARD class (and drug where relevant)

**clinical category and phenotype**  
→ based on rules for the detected markers

# Example genome summary report

*Klebsiella pneumoniae* SGH10 (ampR)

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	combo rules	organism
ciprofloxacin	fluoroquinolone antibiotic	S	wildtype	moderate	-	-	oqxB19;oqxA	KPN0002;KPN0003	-	s_Klebsiella pneumoniae
-	penicillin beta-lactam	R	wildtype	high	blaSHV-11	-	-	KPN0001	-	s_Klebsiella pneumoniae
fosfomycin	phosphonic acid antibiotic	S	wildtype	moderate	-	-	fosA	KPN0004	-	s_Klebsiella pneumoniae
-	antibiotic efflux	-	-	-	-	emrD	-	-	-	s_Klebsiella pneumoniae



Markers annotated by NCBI as Class='EFFLUX' will be printed to their own row with class '**antibiotic efflux**' (the corresponding CARD term).

Markers not assigned to a class (e.g. partial hit, or rule maps to no drug/class) will be printed to their own row '**unassigned markers**'

# Example genome summary report

*Klebsiella pneumoniae* (ampR) + blaCTX-M-15

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	combo rules	organism
ciprofloxacin	fluoroquinolone antibiotic	S	wildtype	moderate	-	-	oqx B19; oqx A	KPN0002; KPN0003	-	s_Klebsiella pneumoniae
-	penicillin beta-lactam	R	wildtype	high	blaSHV-11	-	-	KPN0001	-	s_Klebsiella pneumoniae
fosfomycin	phosphonic acid antibiotic	S	wildtype	moderate	-	-	fosA	KPN0004	-	s_Klebsiella pneumoniae
-	third-generation cephalosporin	R	nonwildtype	-	-	blaCTX-M-15	-	-	-	s_Klebsiella pneumoniae
-	antibiotic efflux	-	-	-	-	emrD	-	-	-	s_Klebsiella pneumoniae

--no-rule-interpretation: 'nwtR' or 'nwtS'

- Parameter determines how to interpret a marker if it has no rule in the genome summary report
  - phenotype assigned 'nonwildtype'
  - clinical category assigned 'R' or 'S', depending on parameter
- Evidence grade defaults to 'very low' (unless markers with rules support the same call)

# Annotated AMRfinderplus report (example: wildtype *E. coli*)



genotype interpretation

Gene symbol	Subclass	Hierarchy node	Variation type	mutation	ruleID	gene context	drug	drug class	phenotype	clinical category	evidence grade	version	organism
pmrB_Y358N	COLISTIN	pmrB	Protein variant detected	p.Tyr358Asn	ECO0081	core	colistin	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
glpT_E448K	FOSFOMYCIN	glpT	Protein variant detected	p.Glu448Lys	ECO0082	core	fosfomycin	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
acrF	EFFLUX	acrF	Inactivating mutation detected	-		-	-	-	-	-	-	-	-
mdtM	EFFLUX	mdtM	Gene presence detected	-		-	-	-	-	-	-	-	-
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0001	core	-	penicillin beta-lactam	wildtype	S	moderate	0.1.0	s_Escherichia coli

# Annotated AMRfinderplus report



AMRfinderplus



genotype interpretation

blaEC row copied for each rule

Gene symbol	Subclass	Hierarchy node	Variation type	mutation	ruleID	gene context	drug	drug class	phenotype	clinical category	evidence grade	version	organism
pmrB_Y358N	COLISTIN	pmrB	Protein variant detected	p.Tyr358Asn	ECO0081	core	colistin	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
glpT_E448K	FOSFOMYCIN	glpT	Protein variant detected	p.Glu448Lys	ECO0082	core	fosfomycin	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
acrF	EFFLUX	acrF	Inactivating mutation detected	-		-	-	-	-	-	-	-	-
mdtM	EFFLUX	mdtM	Gene presence detected	-		-	-	-	-	-	-	-	-
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0001	core	-	penicillin beta-lactam	wildtype	S	moderate	0.1.0	s_Escherichia coli
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0002	core	ampicillin	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0005	core	-	first-generation cephalosporin	wildtype	S	moderate	0.1.0	s_Escherichia coli
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0009	core	cefuroxime	-	wildtype	S	moderate	0.1.0	s_Escherichia coli
blaEC	BETA-LACTAM	blaEC	Gene presence detected	-	ECO0010	core	cefuroxime	-	wildtype	I	moderate	0.1.0	s_Escherichia coli

# Example genome summary report

*E. coli* SAMN26308439 (sensitive to all drugs on BD Phoenix panel)

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	organism
imipenem	carbapenem	S	wildtype	high	-	-	blaEC	ECO0016;ECO0017	s_Escherichia coli
-	carbapenem	S	wildtype	high	-	-	blaEC	ECO0016	s_Escherichia coli
cefazolin	first-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0005;ECO0006	s_Escherichia coli
-	first-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0005	s_Escherichia coli
aztreonam	monobactam	S	wildtype	high	-	-	blaEC	ECO0018	s_Escherichia coli
-	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0001	s_Escherichia coli
amoxicillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0003	s_Escherichia coli
ampicillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0002	s_Escherichia coli
piperacillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0004	s_Escherichia coli
fosfomycin	phosphonic acid antibiotic	S	wildtype	moderate	-	-	glpT:p.Glu448Lys	ECO0082	s_Escherichia coli
colistin	polymyxin antibiotic	S	wildtype	moderate	-	-	pmrB:p.Tyr358Asn	ECO0081	s_Escherichia coli
cefoxitin	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007;ECO0008	s_Escherichia coli
cefuroxime	second-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0007;ECO0009;ECO0010	s_Escherichia coli
-	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007	s_Escherichia coli
cefixime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0015	s_Escherichia coli
cefotaxime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0011;ECO0012	s_Escherichia coli
ceftazidime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0014	s_Escherichia coli
ceftriaxone	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0013	s_Escherichia coli
-	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012	s_Escherichia coli
-	antibiotic efflux	-	-	-	-	-	acrF:-;mdtM	-	s_Escherichia coli

# Example genome summary report

*E. coli* SAMN26308439 (sensitive to all drugs on BD Phoenix panel)

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	organism
imipenem	carbapenem	S	wildtype	high	-	-	blaEC	ECO0016;ECO0017	s_Escherichia coli
-	carbapenem	S	wildtype	high	-	-	blaEC	ECO0016	s_Escherichia coli
cefazolin	first-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0005;ECO0006	s_Escherichia coli
-	first-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0005	s_Escherichia coli
aztreonam	monobactam	S	wildtype	high	-	-	blaEC	ECO0018	s_Escherichia coli
-	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0001	s_Escherichia coli
amoxicillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0003	s_Escherichia coli
ampicillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0002	s_Escherichia coli
piperacillin	penicillin beta-lactam	S	wildtype	high	-	-	blaEC	ECO0004	s_Escherichia coli
fosfomycin	phosphonic acid antibiotic	S	wildtype	moderate	-	-	glpT:p.Glu448Lys	ECO0082	s_Escherichia coli
colistin	polymyxin antibiotic	S	wildtype	moderate	-	-	pmrB:p.Tyr358Asn	ECO0081	s_Escherichia coli
cefoxitin	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007;ECO0008	s_Escherichia coli
cefuroxime	second-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0007;ECO0009;ECO0010	s_Escherichia coli
-	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007	s_Escherichia coli
cefixime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0015	s_Escherichia coli
cefotaxime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0011;ECO0012	s_Escherichia coli
ceftazidime	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0014	s_Escherichia coli
ceftriaxone	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012;ECO0013	s_Escherichia coli
-	third-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0012	s_Escherichia coli
-	antibiotic efflux	-	-	-	-	-	acrF:-;mdtM	-	s_Escherichia coli

# Example genome summary report

*E. coli* SAMN26308439 (sensitive to all drugs on BD Phoenix panel)

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	organism
cefoxitin	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007;ECO0008	s_Escherichia coli
cefuroxime	second-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0007;ECO0009; ECO0010	s_Escherichia coli
-	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007	s_Escherichia coli

Cephalosporins <sup>1</sup>	MIC breakpoints (mg/L)			Disk content (µg)	Zone diameter breakpoints (mm)		
	S ≤	R >	ATU		S ≥	R <	ATU
Cefuroxime iv, <i>E. coli</i> , <i>Klebsiella</i> spp. (except <i>K. aerogenes</i> ), <i>Raoultella</i> spp. and <i>P. mirabilis</i>	0.001	8		30	50	19	
Cefuroxime oral (uncomplicated UTI only), <i>E. coli</i> , <i>Klebsiella</i> spp. (except <i>K. aerogenes</i> ), <i>Raoultella</i> spp. and <i>P. mirabilis</i>	8	8		30	19	19	

*E. coli* group defined separate rules for blaEC

- Cefuroxime => wt I
- 2GC => wt S

Is this needed?

- Oral: breakpoint = ECOFF
- IV: S breakpoints forced to MIC≤=0.001 and zone S>=50, to force reporting as 'I' not 'S'

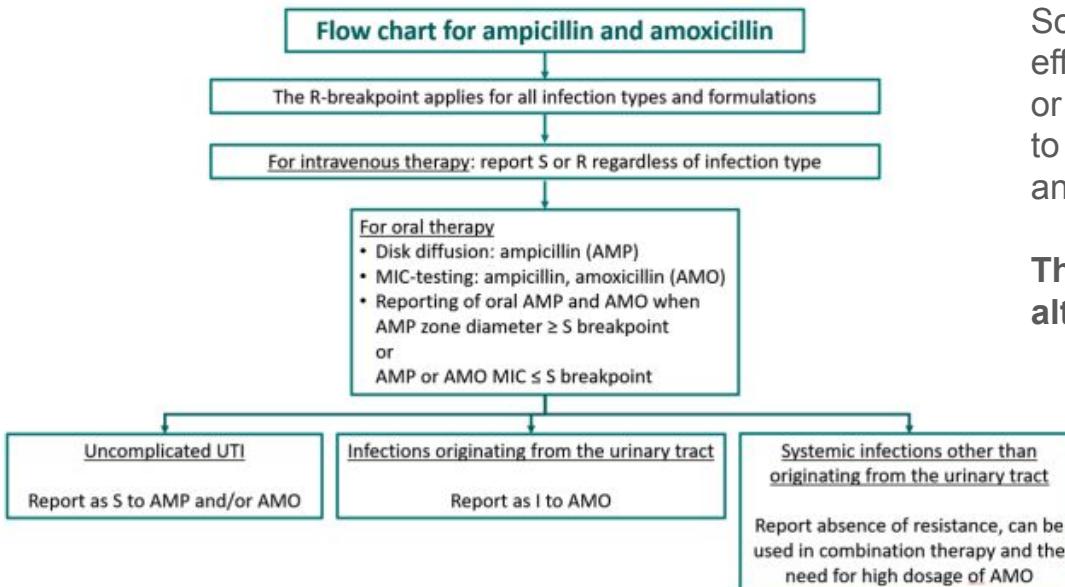
# *E. coli* group defined condition-specific rules

Cephalosporins <sup>1</sup>	MIC breakpoints (mg/L)			Disk content (µg)	Zone diameter breakpoints (mm)		
	S ≤	R >	ATU		S ≥	R <	ATU
Cefuroxime iv, <i>E. coli</i> , <i>Klebsiella</i> spp. (except <i>K. aerogenes</i> ), <i>Raoultella</i> spp. and <i>P. mirabilis</i>	0.001	8		30	50	19	
Cefuroxime oral (uncomplicated UTI only), <i>E. coli</i> , <i>Klebsiella</i> spp. (except <i>K. aerogenes</i> ), <i>Raoultella</i> spp. and <i>P. mirabilis</i>	8	8		30	19	19	

ruleID	nodeID	variation type	gene context	drug	drug class	phenotype	clinical category	breakpoint condition
ECO0007	blaEC	Gene presence detected	core	-	second-generation cephalosporin	wildtype	S	-
ECO0008	blaEC	Gene presence detected	core	cefoxitin	-	wildtype	S	-
ECO0009	blaEC	Gene presence detected	core	cefuroxime	-	wildtype	S	Oral
ECO0010	blaEC	Gene presence detected	core	cefuroxime	-	wildtype	I	Intravenous

← is this needed?

Penicillins	MIC breakpoints (mg/L)			Disk content ( $\mu$ g)	Zone diameter breakpoints (mm)		
	S ≤	R >	ATU		S ≥	R <	ATU
Amoxicillin iv <sup>1</sup>	8	8		-	Note <sup>B</sup>	Note <sup>B</sup>	
Amoxicillin oral (infections originating from the urinary tract) <sup>1</sup>	0.001	8		-	Note <sup>C</sup>	Note <sup>C</sup>	
Amoxicillin oral (uncomplicated UTI only) <sup>1</sup>	8	8		-	Note <sup>B</sup>	Note <sup>B</sup>	
Amoxicillin oral (other indications) <sup>1</sup>	(0.001) <sup>3</sup>	(8) <sup>3</sup>		-	Note <sup>D,E</sup>	Note <sup>D,E</sup>	

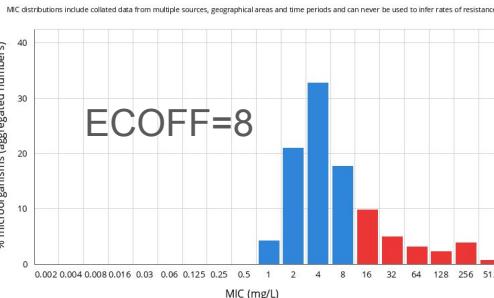


So the 0.001 breakpoint for oral administration is effectively an expert rule to report all isolates as 'I' or 'R' (never 'S'), except for uncomplicated UTI... to avoid suggesting use of oral amoxicillin for anything other than uncomplicated UTI.

**There is no value in defining rules using this alternative S breakpoint.**

Penicillins	MIC breakpoints (mg/L)			Disk content (µg)	Zone diameter breakpoints (mm)		
	S ≤	R >	ATU		S ≥	R <	ATU
Amoxicillin-clavulanic acid iv <sup>1</sup>	8 <sup>4</sup>	8 <sup>4</sup>		20-10	19 <sup>A</sup>	19 <sup>A</sup>	19-20
Amoxicillin-clavulanic acid oral (infections originating from the urinary tract) <sup>1</sup>	0.001 <sup>4</sup>	8 <sup>4</sup>		20-10	50 <sup>A</sup>	19 <sup>A</sup>	19-20
Amoxicillin-clavulanic acid oral (uncomplicated UTI only) <sup>1</sup>	32 <sup>4</sup>	32 <sup>4</sup>		20-10	16 <sup>A</sup>	16 <sup>A</sup>	
Amoxicillin-clavulanic acid oral (other indications) <sup>1</sup>	(0.001) <sup>3,4</sup>	(8) <sup>3,4</sup>		20-10	(50) <sup>A,D</sup>	(19) <sup>A,D</sup>	19-20

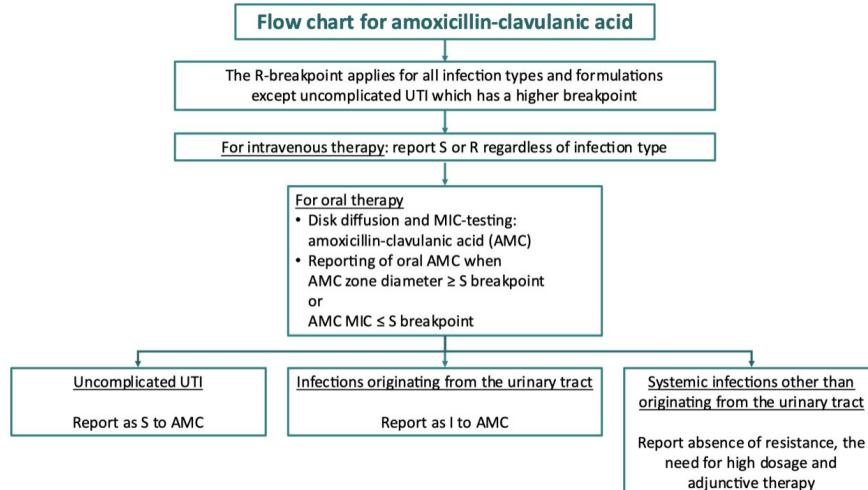
Amoxicillin-clavulanic acid (fixed) / Escherichia coli  
International MIC distribution - Reference database 2025-11-27  
Based on aggregated distributions



- IV breakpoint = same as ECOFF
- Oral, uncomplicated UTI = breakpoint 32
- Oral, other = same 'R' but report rest as 'I' not 'S'

So if we define rules using the Oral, uncomplicated UTI breakpoint (32 mg/L) and ECOFF, the resulting S/I/R and WT/NWT calls could be used to interpret wrt all of these rules, either in AMRrules itself or (preferred) in downstream clinical reporting logic to apply these as expert rules.

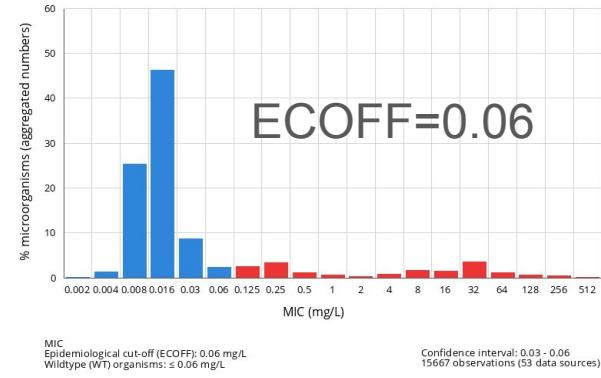
(CLSI breakpoints: MIC S=8 mg/L, R=32 mg/L; disk zone S=18 mm, R=13 mm)



MIC distributions include collated data from multiple sources, geographical areas and time periods and can never be used to infer rates of resistance

## Condition-specific breakpoints for *E. coli*

Fluoroquinolones	MIC breakpoints (mg/L)			Disk content (µg)	Zone diameter breakpoints (mm)		
	S ≤	R >	ATU		S ≥	R <	ATU
Ciprofloxacin, <i>Salmonella</i> spp. <sup>1</sup>	0.06	0.06			Note <sup>A</sup>	Note <sup>A</sup>	
Ciprofloxacin (indications other than meningitis)	0.25	0.5	0.5	5	25	22	22-24
Ciprofloxacin (meningitis) <sup>2</sup>	0.125	0.125			Note <sup>B</sup>	Note <sup>B</sup>	



**Guidance note 2/B: “In meningitis, where all fluoroquinolone resistance mechanisms must be excluded, either perform an MIC test, or infer susceptibility from the pefloxacin 5 µg screening test.”**

- So the purpose of the lower meningitis breakpoint is to try to identify presence of any resistance mechanism, to avoid recommending use of the drug in these circumstances... but with genome data, genotype interpretation can be done directly rather than defining rules wrt an MIC
- We essentially cover this already with the call of wildtype/nonwildtype vs ECOFF ( $\leq 0.06$ )
- **So there is no need to define rules separately to accommodate this breakpoint**
- Instead, it could be built into downstream clinical reporting logic which could flag the presence of any NWT call for ciprofloxacin as ruling out meningitis

# Example genome summary report

*E. coli* SAMN26308439 (sensitive to all drugs on BD Phoenix panel)

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs	organism
cefoxitin	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007;ECO0008	s_Escherichia coli
cefuroxime	second-generation cephalosporin	I	wildtype	high	blaEC	-	blaEC	ECO0007;ECO0009; ECO0010	s_Escherichia coli
-	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007	s_Escherichia coli

## Solution:

1. Do not define condition-specific rules when the purpose is to enforce reporting. This should be handled downstream as a reporting rule.
2. If there is a genuine need to define multiple genotype interpretation rules to correspond to condition-specific breakpoints, AMRrules engine will need to be updated to accommodate this.

**Do we need condition-specific rules?**

**If we do need condition-specific rules, how should this be reflected in the report?**

## If we do need condition-specific rules, how should this be reflected in the report?

rules

ruleID	nodeID	variation type	gene context	drug	drug class	phenotype	clinical category	breakpoint condition
ECO0007	blaEC	Gene presence detected	core	-	second-generation cephalosporin	wildtype	S	-
ECO0008	blaEC	Gene presence detected	core	cefoxitin	-	wildtype	S	-
ECO0009	blaEC	Gene presence detected	core	cefuroxime	-	wildtype	S	Oral
ECO0010	blaEC	Gene presence detected	core	cefuroxime	-	wildtype	I	Intravenous

report

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleID
-	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0007
cefoxitin	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0008
cefuroxime <i>(Oral)</i>	second-generation cephalosporin	S	wildtype	high	-	-	blaEC	ECO0009
cefuroxime <i>(Intravenous)</i>	second-generation cephalosporin	I	wildtype	high	blaEC	-	-	ECO0010

# How to interpret partial hits, to genes with a detection rule?

Gene symbol	Method	Hierarchy node	variation type	gene	mutation	ruleID
aac(3)-la	BLASTX	aac(3)-la	Gene presence detected	aac(3)-la	-	-
aadA1	EXACTX	aadA1	Gene presence detected	aadA1	-	-
sul1	EXACTX	sul1	Gene presence detected	sul1	-	-
parC_S84L	POINTX	parC	Protein variant detected	parC	p.Ser84Leu	-
adeC	PARTIAL_CONTIG_ENDX	adeC	Inactivating mutation detected	adeC	-	-
aph(6)-Id	EXACTX	aph(6)-Id	Gene presence detected	aph(6)-Id	-	-
aph(3")-lb	EXACTX	aph(3")-lb	Gene presence detected	aph(3")-lb	-	-
gyrA_S81L	POINTX	gyrA	Protein variant detected	gyrA	p.Ser81Leu	ACI0162
aph(3')-la	EXACTX	aph(3')-la	Gene presence detected	aph(3')-la	-	-
tet(B)	BLASTX	tet(B)	Gene presence detected	tet(B)	-	-
sul2	EXACTX	sul2	Gene presence detected	sul2	-	-
amvA	PARTIAL_CONTIG_ENDX	amvA	Inactivating mutation detected	amvA	-	-
blaADC	PARTIALX	blaADC	Inactivating mutation detected	blaADC	-	-
blaOXA	BLASTX	blaOXA-51_fam	Gene presence detected	blaOXA	-	ACI0146
ant(3")-lla	PARTIALX	ant(3")-lla	Inactivating mutation detected	ant(3")-lla	-	-

# How to interpret partial hits, to genes with a detection rule?

*rules for these genes*

ruleID	gene	variation type	gene context	drug	drug class	phenotype	clinical category
ACI0152	blaADC	Gene presence detected	core	-	third-generation cephalosporin	wildtype	R
ACI0153	blaADC	Gene presence detected	core	ceftazidime	-	wildtype	R
ACI0154	blaADC	Gene presence detected	core	cefotaxime	-	wildtype	R
ACI0158	ant(3")-II	Gene presence detected	core	gentamicin	-	wildtype	S
ACI0159	ant(3")-II	Gene presence detected	core	tobramycin	-	wildtype	S
ACI0165	ant(3")-II	Gene presence detected	core	amikacin	-	wildtype	S
ACI0166	ant(3")-II	Gene presence detected	core	spectinomycin	-	wildtype	R
ACI0167	ant(3")-II	Gene presence detected	core	streptomycin	-	wildtype	R
ACI0160	amvA	Gene presence detected	core	-	-	wildtype	S

**Propose:** (i) report as markers (S) if there is a rule to interpret detection as wt S, and no other markers for the drug; otherwise report as unassigned markers.

(ii) Subgroups define rules for inactivation of core genes where there is evidence.

rules:

ACI0158	ant(3")-II	Gene presence detected	core	gentamicin	wildtype	S
ACI0159	ant(3")-II	Gene presence detected	core	tobramycin	wildtype	S
ACI0165	ant(3")-II	Gene presence detected	core	amikacin	wildtype	S
ACI0160	amvA	Gene presence detected	core	-	wildtype	S

Thoughts?

drug	drug class	clinical category	phenotype	evidence grade	markers (non-S)	markers (no rule)	markers (S)	ruleIDs
gentamicin	aminoglycoside antibiotic	R	nonwildtype	very low	-	aac(3')-la	-	-
amikacin	aminoglycoside antibiotic	S	wildtype	high	-	-	ant(3")-Ila:-	ACI0165
tobramycin	aminoglycoside antibiotic	S	wildtype	high	-	-	ant(3")-Ila:-	ACI0159
kanamycin A	aminoglycoside antibiotic	R	nonwildtype	very low	-	aph(3')-la	-	-
streptomycin	aminoglycoside antibiotic	R	nonwildtype	very low	-	aadA1;aph(6)-Id;aph(3")-Ib	-	-
-	carbapenem	S	wildtype	moderate	-	-	blaOXA	ACI0146
-	fluoroquinolone antibiotic	R	nonwildtype	high	gyrA:p.Ser81Leu	parC:p.Ser84Leu	-	ACI0162
-	sulfonamide antibiotic	R	nonwildtype	very low	-	sul1;sul2	-	-
-	tetracycline antibiotic	R	nonwildtype	very low	-	tet(B)	-	-
-	unassigned markers	S	nonwildtype	very low	-	adeC:-;amvA:-; blaADC:-	-	-

## AMRrules package - steps to v1.0 release

1. Incorporate additional core gene rules submitted
  - *Bordetella, Shewanella, Burkholderia pseudomallei, Legionella, Proteus mirabilis, Achromobacter xylosoxidans, Neisseria meningitidis, Klebsiella oxytoca.*
2. Accept hAMRonized AMRfinderplus input (stretch goal: CARD)
3. Call on working group to beta-test on supported organisms
  - Test genomes and output files will be distributed for review
  - WG members will be asked to test software on own genomes
  - Provide feedback on documentation and output files

AIM: Release by early March (ahead of Wellcome AMR Conference)

# Please provide feedback on the AMRrules interpretation engine

- Look at test outputs in AMRrules repo tests/ (development branch)
  - [https://github.com/AMRverse/AMRrules/tree/genome\\_summary\\_report\\_dev/tests/data/example\\_output](https://github.com/AMRverse/AMRrules/tree/genome_summary_report_dev/tests/data/example_output)
- Test AMRrules package (development branch) on your own genomes
  - [https://amrrules.readthedocs.io/en/genome\\_summary\\_report\\_dev/installation.html](https://amrrules.readthedocs.io/en/genome_summary_report_dev/installation.html)
- Examples where condition-specific breakpoints needeed?
- Thoughts about interpretation of ‘partial’ hits to
  - core genes?
  - acquired genes?

## How to share feedback

- Post an issue: <https://github.com/AMRverse/AMRrules/issues>
- Slack channel #codequeries
- Email esgem.amr@gmail.com

# Agenda

1. **Online resources:** where to find things
2. **AMRrules Python package:** applying rules to interpret genotypes
3. **AMRrulemakeR package:** defining rules from geno-pheno data
4. **2026 planning**

- Manuscript plans
- Meeting schedule
- Events

# AMRgen R package

<https://github.com/AMRverse/AMRgen>

## Import AST results to standard format using AMR classes

```
import_ast("~/Downloads/Ecoli_AST_EBI.csv", format="ebi", interpret_eucast = T, interpret_ecoff = T)  
import_ast("~/Downloads/Ecoli_AST_NCBI.csv", format="ncbi", interpret_eucast = T, interpret_ecoff = T)
```

	id	drug_agent	mic	disk	pheno_eucast	pheno_clsi	ecoff	guideline	method	source	pheno_provided	spp_pheno
	<chr>	<ab>	<mic>	<dsk>	<sir>	<sir>	<sir>	<chr>	<chr>	<chr>	<sir>	<m>
1	SAMN02437318	CAZ	>16	NA	R	R	R	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
2	SAMN02437318	CTX	>32	NA	R	R	S	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
3	SAMN02437318	LVX	>4	NA	R	R	R	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
4	SAMN02437318	MEM	>=16	NA	R	R	R	CLSI	Vitek	33659219	R	B_ACNTB_BMNN
5	SAMN02437318	TCY	>8	NA	NA	R	NA	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
6	SAMN02437318	TGC	1	NA	R	NA	NA	CLSI	Vitek	33659219	S	B_ACNTB_BMNN
7	SAMN02437332	CAZ	>16	NA	R	R	R	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
8	SAMN02437332	CTX	>32	NA	R	R	S	CLSI	Microscan	33659219	R	B_ACNTB_BMNN
9	SAMN02437332	LVX	4	NA	R	I	R	CLSI	Microscan	33659219	I	B_ACNTB_BMNN
10	SAMN02437332	MEM	>=16	NA	R	R	R	CLSI	Vitek	33659219	R	B_ACNTB_BMNN

AMR package classes

source to track unique datasets

# AMRgen R package

<https://github.com/AMRverse/AMRgen>

```
geno <- import_amrfp("amrfinderplus.tsv", sample_col = "Name")
```

## Input

Name	'Gene symbol'	'Element type'	'Element subtype'	Class	Subclass	Method
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	002_S03 aac(3)-IIe	AMR	AMR	AMINOGLYCOSIDE	GENTAMICIN	EXACTX
2	002_S03 aph(3')-Ib	AMR	AMR	AMINOGLYCOSIDE	STREPTOMYCIN	EXACTX
3	002_S03 aph(6)-Id	AMR	AMR	AMINOGLYCOSIDE	STREPTOMYCIN	EXACTX
4	002_S03 aadA2	AMR	AMR	AMINOGLYCOSIDE	STREPTOMYCIN	EXACTX
5	002_S03 aadA1	AMR	AMR	AMINOGLYCOSIDE	STREPTOMYCIN	PARTI...
6	002_S03 oqx8	AMR	AMR	PHENICOL/QUINOLONE	PHENICOL/QU...	BLASTX
7	002_S03 oqxA	AMR	AMR	PHENICOL/QUINOLONE	PHENICOL/QU...	EXACTX
8	003_S08 rmtB1	AMR	AMR	AMINOGLYCOSIDE	AMINOGLYCOS...	ALLEL...
9	003_S08 aac(3)-IID	AMR	AMR	AMINOGLYCOSIDE	GENTAMICIN	EXACTX
10	003_S08 aph(3')-Ia	AMR	AMR	AMINOGLYCOSIDE	KANAMYCIN	EXACTX

## Output

Name	gene	mutation	node	'variation type'	marker	marker.label	drug_agent	drug_class	'Gene symbol'
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	002_S03 ompK36	Asp135AspGlyAsp	ompK36	Protein variant detected	ompK36_D135DGD	ompK36:Asp135AspGlyAsp	NA	Carbapenems	ompK36_D135D...
2	002_S03 gyrA	Ser83Ile	gyrA	Protein variant detected	gyrA_S83I	gyrA:Ser83Ile	NA	Quinolones	gyrA_S83I
3	002_S03 parC	Ser80Ile	parC	Protein variant detected	parC_S80I	parC:Ser80Ile	NA	Quinolones	parC_S80I
4	002_S03 aac(3)-IIe	NA	aac(3)-IIe	Gene presence detected	aac(3)-IIe	aac(3)-IIe	GEN	Aminoglycosi...	aac(3)-IIe
5	002_S03 aph(3')-Ib	NA	aph(3')-Ib	Gene presence detected	aph(3')-Ib	aph(3')-Ib	STR1	Aminoglycosi...	aph(3')-Ib
6	002_S03 aph(6)-Id	NA	aph(6)-Id	Gene presence detected	aph(6)-Id	aph(6)-Id	STR1	Aminoglycosi...	aph(6)-Id
7	002_S03 aadA2	NA	aadA2	Gene presence detected	aadA2	aadA2	STR1	Aminoglycosi...	aadA2
8	002_S03 aadA1	NA	aadA1	Inactivating mutation detected	aadA1	aadA1:-	STR1	Aminoglycosi...	aadA1
9	002_S03 oqx8	NA	oqx8	Gene presence detected	oqx8	oqx8	NA	Amphenicols	oqx8
10	002_S03 oqxA	NA	oqxA	Gene presence detected	oqxA	oqxA	NA	Amphenicols	oqxA
11	002_S03 oqxB	NA	oqxB	Gene presence detected	oqxB	oqxB	NA	Quinolones	oqxB
12	002_S03 oqxA	NA	oqxA	Gene presence detected	oqxA	oqxA	NA	Quinolones	oqxA

AMRrules syntax for specifying variants

marker.label for analysis (node:mutation)

} copy for  
each class

# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

## Examples

```
library(AMRgen)
library(AMRrulemakeR)
library(tidyverse)

# example data included in AMRrulemakeR package
ecoli_ast_ebi
ecoli_afp_atb

# run quantitative analyses for ciprofloxacin phenotypes vs quinolone marker genotypes, using
cip_analysis <- amrrules_analysis(geno_table=ecoli_afp_atb, pheno_table=ecoli_ast_ebi,
                                    antibiotic="Ciprofloxacin", drug_class_list=c("Quinolones",
                                    sir_col="pheno_eucast", ecoff_col="ecoff",
                                    species="Escherichia coli",
                                    info=ecoli_ast_ebi %>% select(id, source, method))

# check key output plots
cip_analysis$ppv_plot
cip_analysis$ppv_plot_all # this includes data with S/I/R interpretations from EBI but no raw
cip_analysis$logistic_plot # note this is only used if the marker is not found solo, to support
cip_analysis$upset_mic_plot
```

# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

## Inbuilt phenotype test data (from EBI AMR portal)

```
> ecoli_ast_ebi
# A tibble: 98,236 × 47
  id      drug_agent    mic   disk pheno_eucast pheno_clsi ecoff guideline method source pheno_provided
  <chr>    <ab>     <mic> <dsk> <sir>          <sir>    <sir> <chr>    <chr> <chr> <chr> <chr>
  1 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  2 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  3 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  4 SAMN13... AMP       <=2    NA    S           S       S  EUCAST  BD Ph... 32205... S
  5 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  6 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  7 SAMN13... AMP        >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
  8 SAMN13... AMP       <=2    NA    S           S       S  EUCAST  BD Ph... 32205... S
  9 SAMN13... AMP       <=2    NA    S           S       S  EUCAST  BD Ph... 32205... S
 10 SAMN13... AMP       >8    NA    R           NI       R  EUCAST  BD Ph... 32205... R
# i 98,226 more rows
```

# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

Inbuilt genotype test data (from Allthebacteria, samples matching phenotype test data)

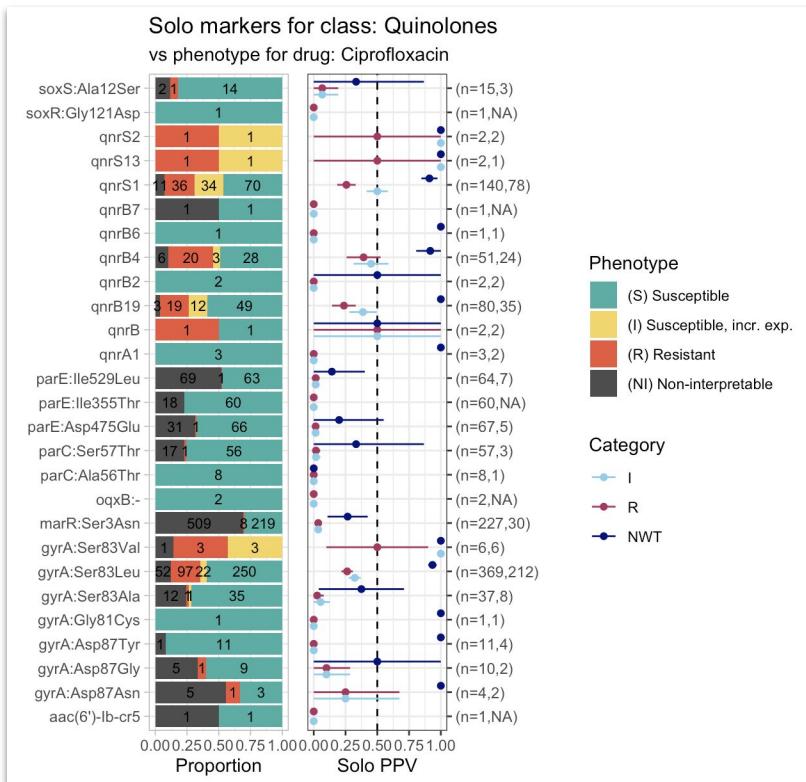
```
> ecoli_afp_atb
# A tibble: 139,758 × 36
  Name      gene mutation node `variation type` marker marker.label drug_agent drug_class status
  <chr>     <chr> <chr>   <chr> <chr>       <chr> <chr>       <ab>    <chr>     <chr>
1 SAMN03177641 blaC... NA        blaC... Gene presence d... blaCM... blaCMY-2      NA        Cephalosp... PASS
2 SAMN03177641 blaEC NA        blaEC Gene presence d... blaEC  blaEC          NA        Beta-lact... PASS
3 SAMN03177641 aph(... NA        aph(... Gene presence d... aph(6... aph(6)-Id      STR1      Aminoglyc... PASS
4 SAMN03177641 aph(... NA        aph(... Gene presence d... aph(3... aph(3')-Ib      STR1      Aminoglyc... PASS
5 SAMN03177641 sul2  NA        sul2  Gene presence d... sul2    sul2          SSS       Sulfonami... PASS
6 SAMN03177641 blaT... NA        blaT... Gene presence d... blaTE... blaTEM-1      NA        Beta-lact... PASS
7 SAMN10101145 dfrA1 NA        dfrA1 Gene presence d... dfrA1  dfrA1          NA        Trimethop... PASS
8 SAMN10101145 sat2  NA        sat2... Gene presence d... sat2   sat2_fam      STR       Aminoglyc... PASS
9 SAMN10101145 aadA1 NA        aadA1 Gene presence d... aadA1  aadA1          STR1      Aminoglyc... PASS
10 SAMN10101145 mph(... NA       mph(... Gene presence d... mph(A) mph(A)        AZM      Macrolide... PASS
# i 139,748 more rows
```

# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

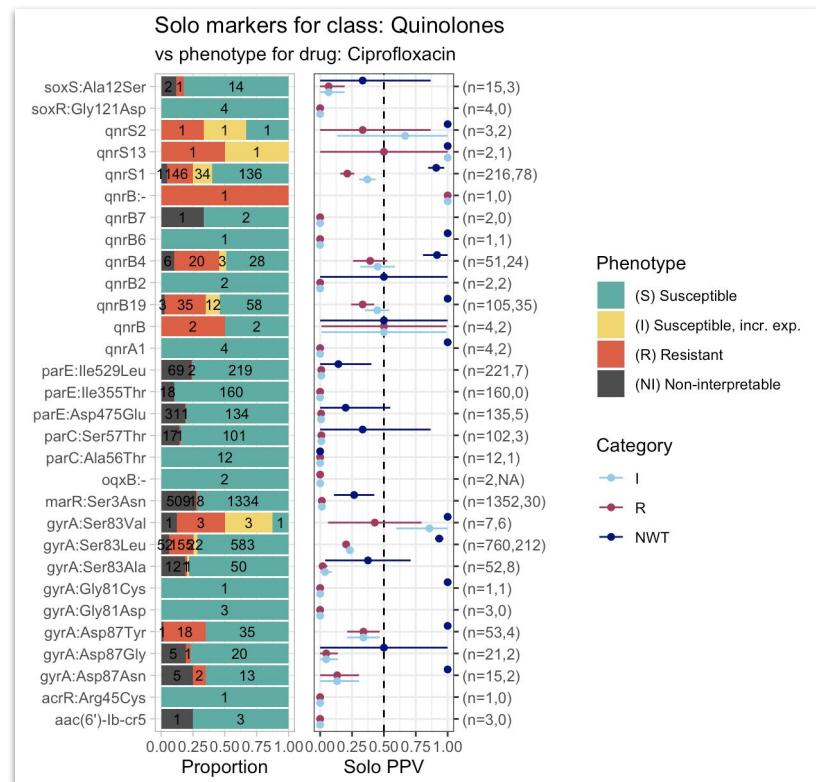
## Primary data: samples with MIC/disk measures

> `cip_analysis$ppv_plot`



## Extended data: all samples with S/I/R calls

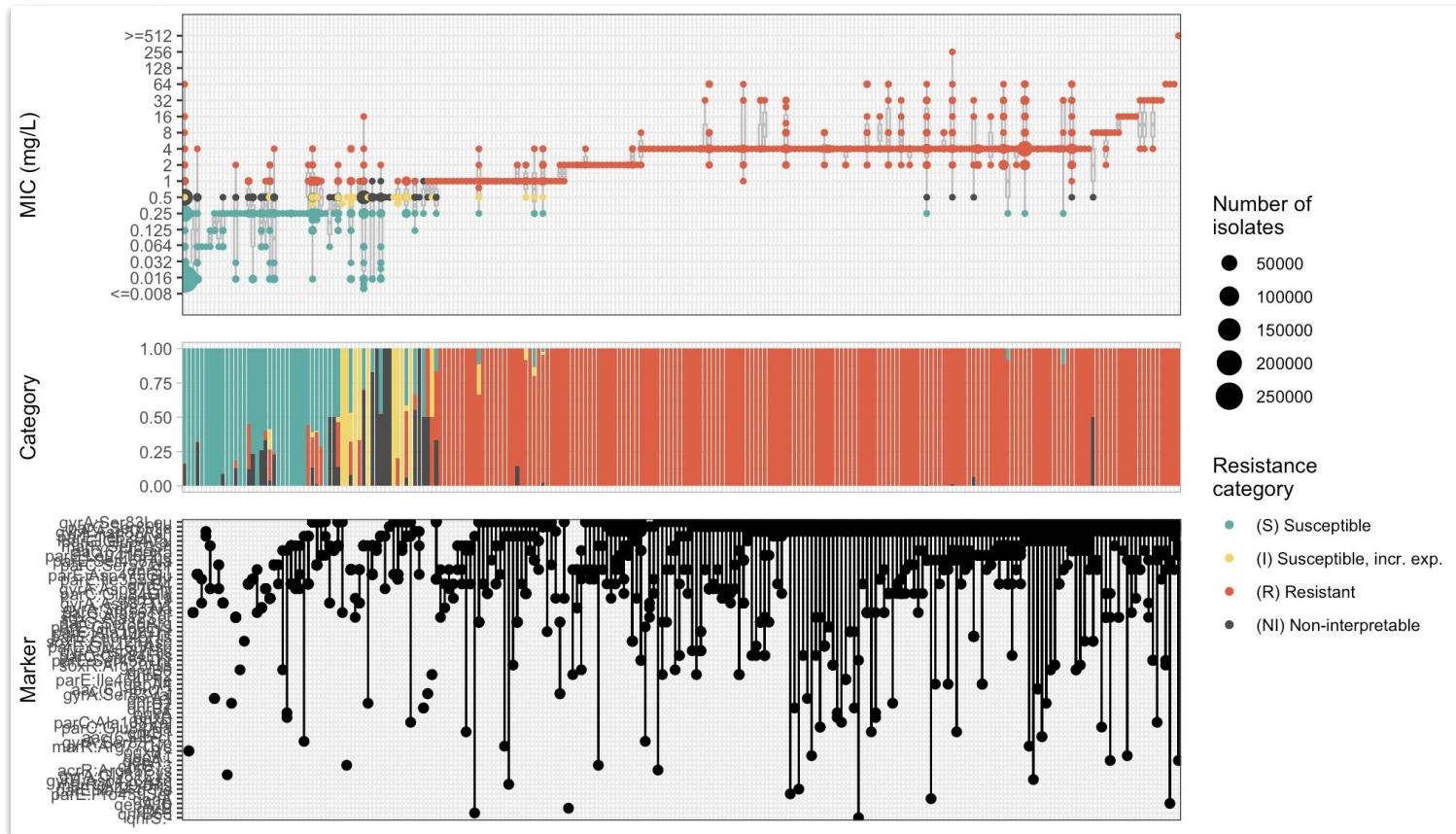
> `cip_analysis$ppv_plot_all`



# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

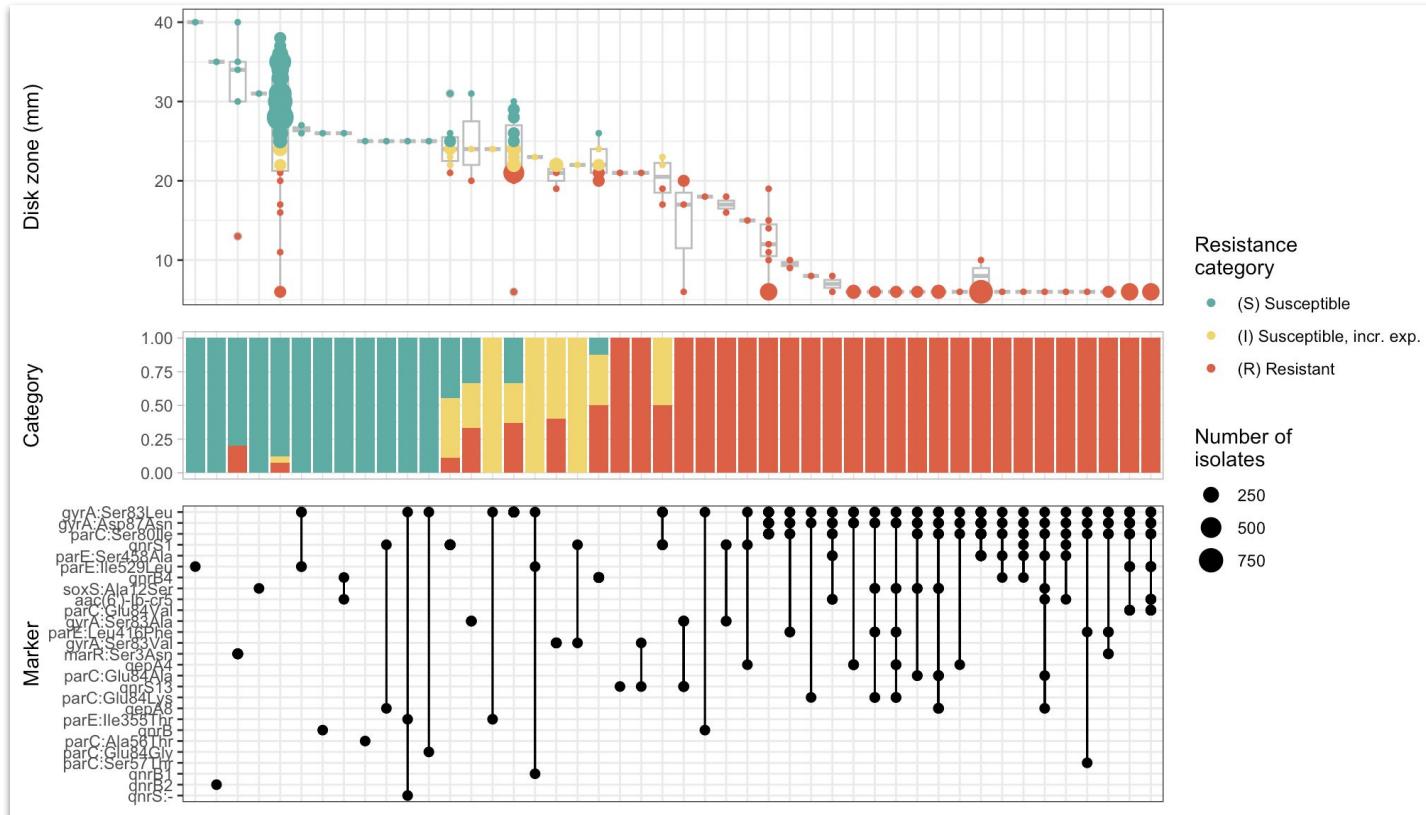
```
> cip_analysis$upset_mic_plot
```



# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

> cip\_analysis\$upset\_disk\_plot



# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

```
# save analysis tables and plots, and generate rules using the Non-meningitis breakpoint, save  
# then use the rules to predicted phenotypes from genotypes and compare to the observed phenot  
cip_rules <- amrrules_save(cip_analysis, bp_site="Non-meningitis",  
                           dir_path="amrrules", file_prefix="Ciprofloxacin")  
  
# alternatively, call makerules directly on the analysis object without saving outputs or runn  
cip_rules <- makerules(cip_analysis, bp_site="Non-meningitis")  
  
# view the proposed rules, in AMRrules specification format, with quantitative fields added  
view(cip_rules$rules)  
  
# manually apply rules to interpret quinolone marker genotypes  
cip_test <- test_rules_amrfp(ecoli_afp_atb %>% filter(drug_class %in% c("Quinolones")),  
                           rules=cip_rules$rules, species="Escherichia coli")  
  
# compare these to the input phenotypes  
cip_test %>% left_join(ecoli_ast_ebi, join_by("Name"=="id")) %>% count(category,pheno_eucast)  
  
# make some plots to view predictions vs raw assay values from different methods, and  
# explore positive predictive value of the overall ruleset including stratified by method  
compare_pred <- compare_interpretations(pred=cip_test, obs=ecoli_ast_ebi,  
                                         antibiotic="Ciprofloxacin",  
                                         sir_col="pheno_eucast", ecoff_col="ecoff",  
                                         var="method")
```

# AMRrulemakeR package

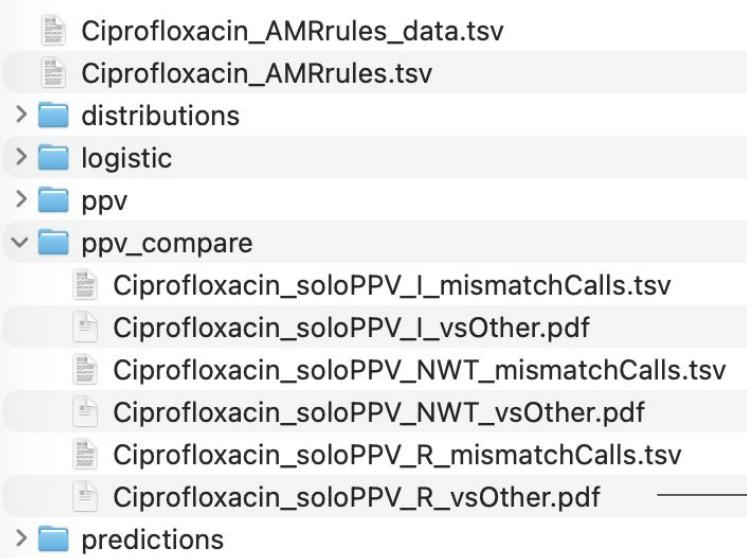
<https://github.com/AMRverse/AMRrulemakeR>

CSV	Ciprofloxacin_AMRrules_data.tsv
CSV	Ciprofloxacin_AMRrules.tsv
> F	distributions
> F	logistic
CSV	Ciprofloxacin_logistic_NWT.tsv
CSV	Ciprofloxacin_logistic_R.tsv
PDF	Ciprofloxacin_logistic.pdf
> F	ppv
CSV	Ciprofloxacin_disk_summary.tsv
PDF	Ciprofloxacin_disk_upset.pdf
CSV	Ciprofloxacin_MIC_summary.tsv
PDF	Ciprofloxacin_MIC_upset.pdf
PDF	Ciprofloxacin_soloPPV_ext.pdf
CSV	Ciprofloxacin_soloPPV_ext.tsv
PDF	Ciprofloxacin_soloPPV_logistic.pdf
PDF	Ciprofloxacin_soloPPV.pdf
CSV	Ciprofloxacin_soloPPV.tsv
> F	ppv_compare
> F	predictions

PPV plots  
Upset plots  
Logistic regression plots

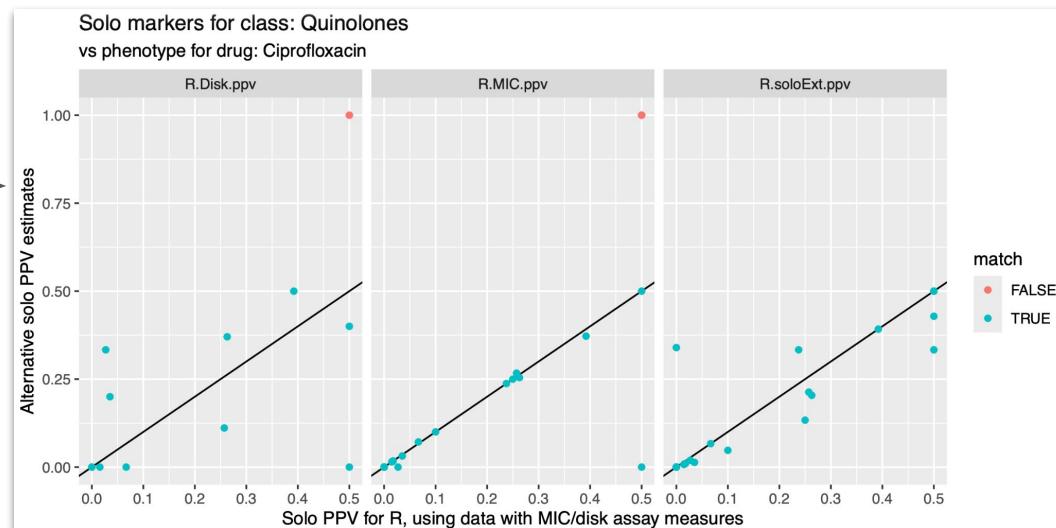
# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>



## Compare S/I/R calls based on primary solo PPV, vs:

- MIC data only
- Disk data only
- Extended data (all samples with S/I/R call, with MIC or disk measure or no measure)

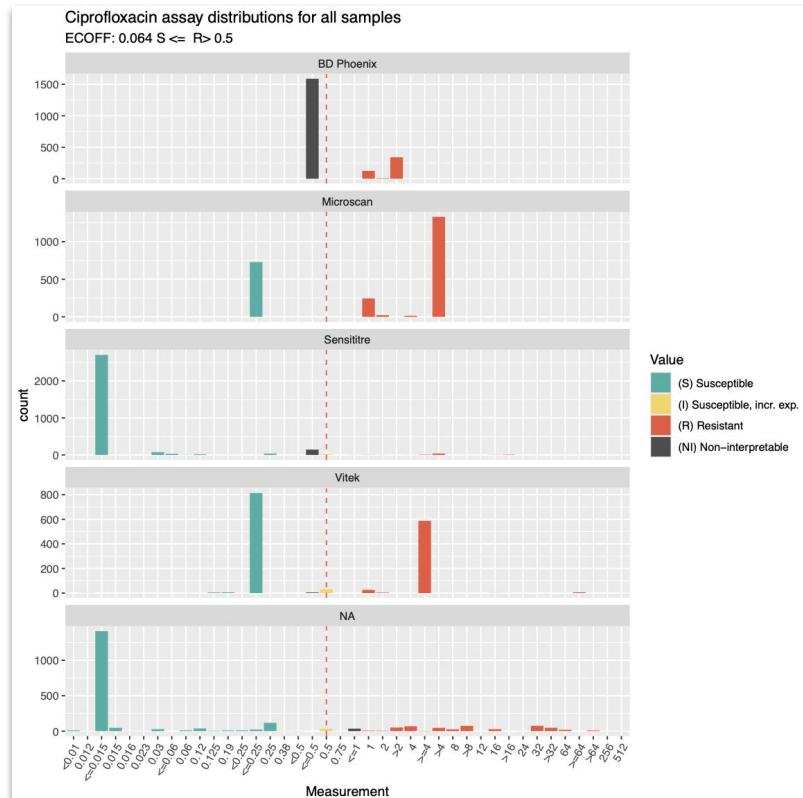


# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

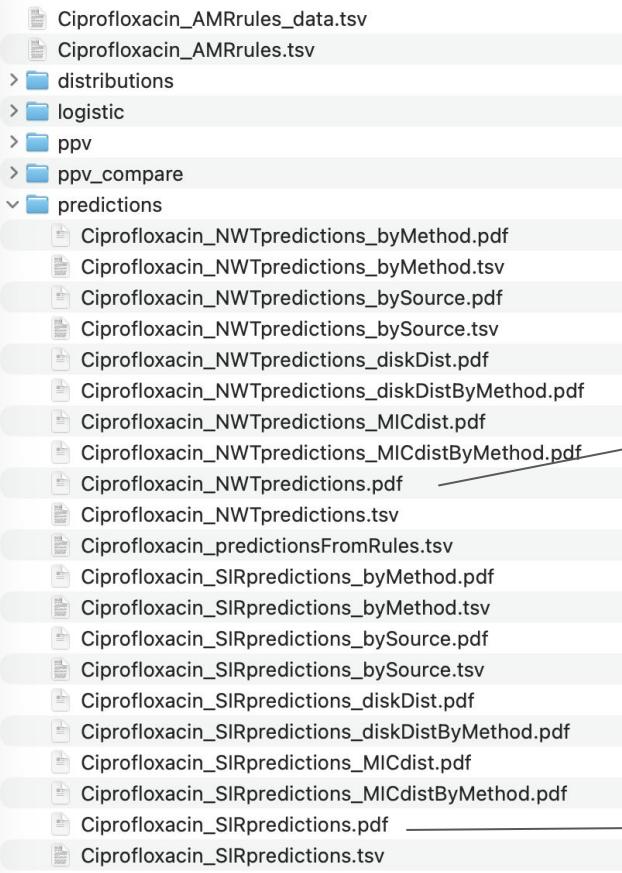
- 📄 Ciprofloxacin\_AMRrules\_data.tsv
- 📄 Ciprofloxacin\_AMRrules.tsv
- 📁 distributions
  - 📄 Ciprofloxacin\_inputDisk\_byNWT\_byMethod.pdf
  - 📄 Ciprofloxacin\_inputDisk\_bySIR\_byMethod.pdf
  - 📄 Ciprofloxacin\_inputDisk\_bySIR.pdf
  - 📄 Ciprofloxacin\_inputMIC\_byNWT\_byMethod.pdf
  - 📄 Ciprofloxacin\_inputMIC\_bySIR\_byMethod.pdf
  - 📄 Ciprofloxacin\_inputMIC\_bySIR.pdf
  - 📄 Ciprofloxacin\_reference\_disk\_plot.pdf
  - 📄 Ciprofloxacin\_reference\_mic\_plot.pdf
- > 📁 logistic
- > 📁 ppv
- > 📁 ppv\_compare
- > 📁 predictions

## MIC distribution of input data, by platform



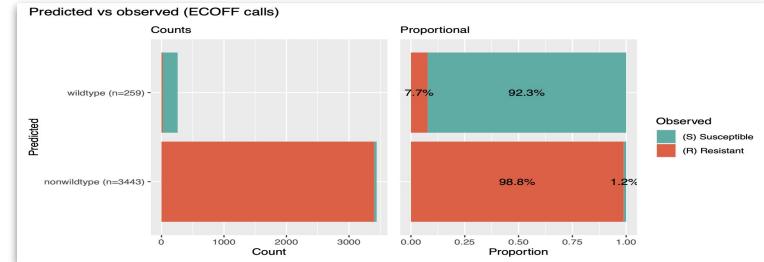
# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

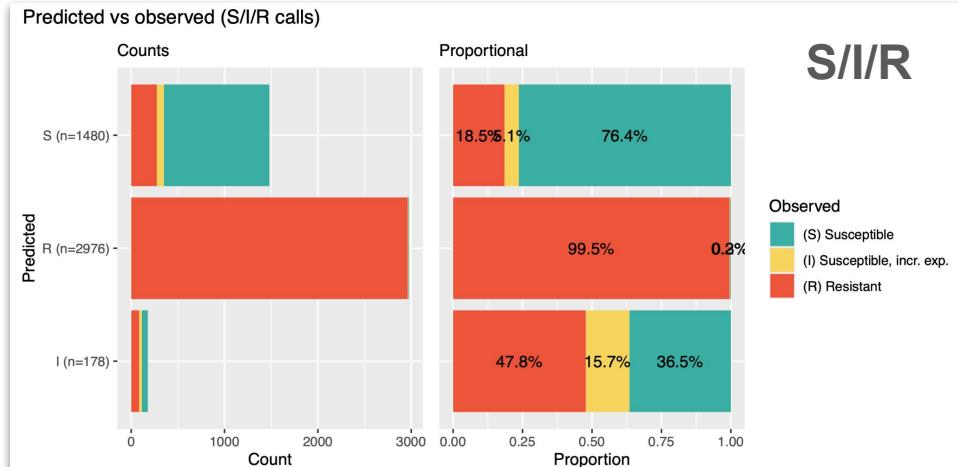


## Positive predictive value of rule-based calls

WT vs NWT



S/I/R

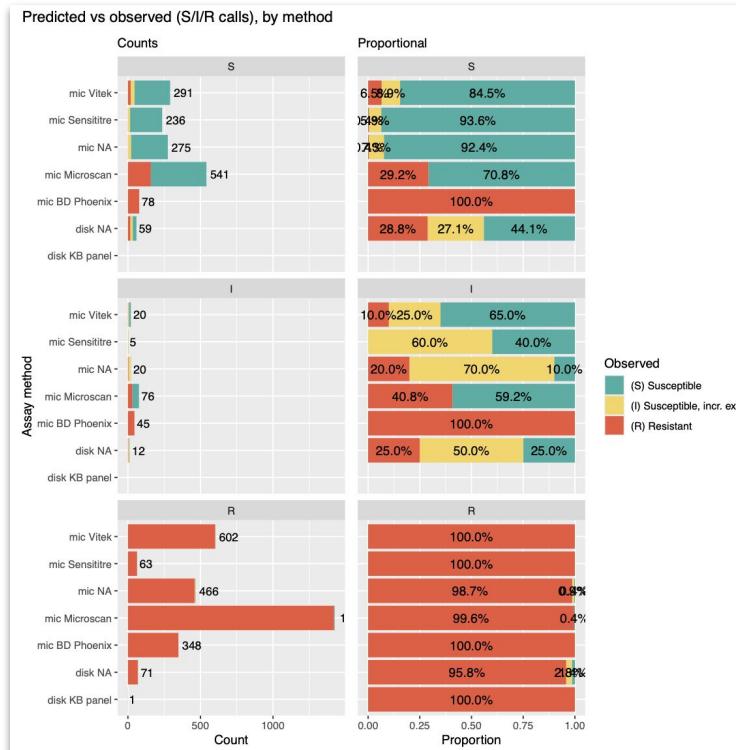


# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

📁 Ciprofloxacin_AMRrules_data.tsv
📄 Ciprofloxacin_AMRrules.tsv
> 📁 distributions
> 📁 logistic
> 📁 ppv
> 📁 ppv_compare
⌄ 📁 predictions
📝 Ciprofloxacin_NWTpredictions_byMethod.pdf
📝 Ciprofloxacin_NWTpredictions_byMethod.tsv
📝 Ciprofloxacin_NWTpredictions_bySource.pdf
📝 Ciprofloxacin_NWTpredictions_bySource.tsv
📝 Ciprofloxacin_NWTpredictions_diskDist.pdf
📝 Ciprofloxacin_NWTpredictions_diskDistByMethod.pdf
📝 Ciprofloxacin_NWTpredictions_MICdist.pdf
📝 Ciprofloxacin_NWTpredictions_MICdistByMethod.pdf
📝 Ciprofloxacin_NWTpredictions.pdf
📝 Ciprofloxacin_NWTpredictions.tsv
📝 Ciprofloxacin_predictionsFromRules.tsv
📝 Ciprofloxacin_SIRpredictions_byMethod.pdf
📝 Ciprofloxacin_SIRpredictions_byMethod.tsv
📝 Ciprofloxacin_SIRpredictions_bySource.pdf
📝 Ciprofloxacin_SIRpredictions_bySource.tsv
📝 Ciprofloxacin_SIRpredictions_diskDist.pdf
📝 Ciprofloxacin_SIRpredictions_diskDistByMethod.pdf
📝 Ciprofloxacin_SIRpredictions_MICdist.pdf
📝 Ciprofloxacin_SIRpredictions_MICdistByMethod.pdf
📝 Ciprofloxacin_SIRpredictions.pdf
📝 Ciprofloxacin_SIRpredictions.tsv

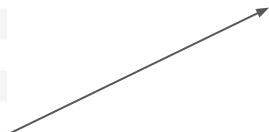
## Positive predictive value of rule-based call stratified by assay type/platform



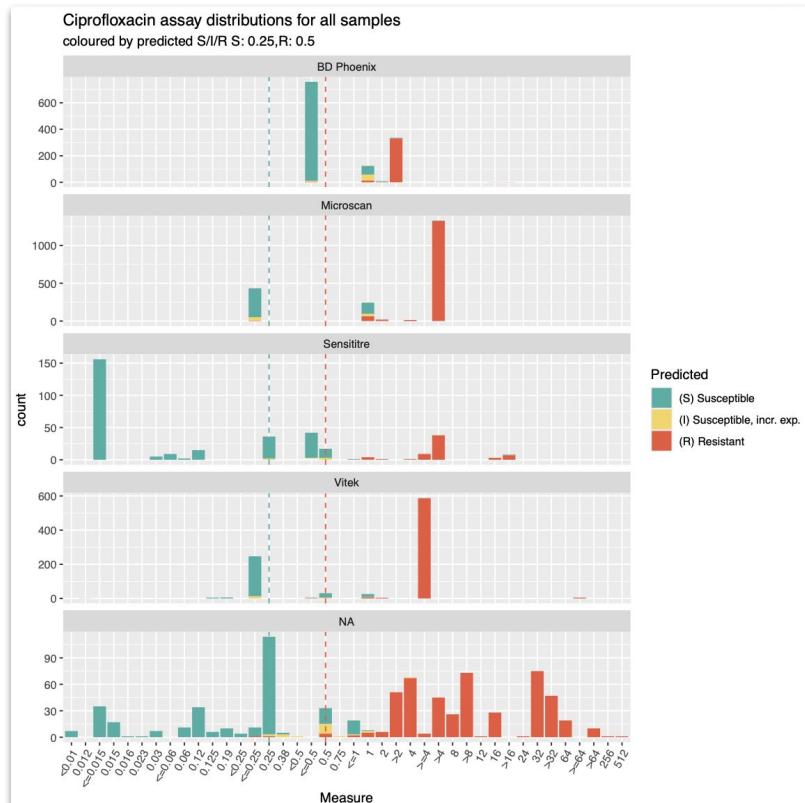
# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

CSV	Ciprofloxacin_AMRrules_data.tsv
CSV	Ciprofloxacin_AMRrules.tsv
> F	distributions
> F	logistic
> F	ppv
> F	ppv_compare
> F	predictions
PDF	Ciprofloxacin_NWTpredictions_byMethod.pdf
CSV	Ciprofloxacin_NWTpredictions_byMethod.tsv
PDF	Ciprofloxacin_NWTpredictions_bySource.pdf
CSV	Ciprofloxacin_NWTpredictions_bySource.tsv
PDF	Ciprofloxacin_NWTpredictions_diskDist.pdf
PDF	Ciprofloxacin_NWTpredictions_diskDistByMethod.pdf
PDF	Ciprofloxacin_NWTpredictions_MICdist.pdf
PDF	Ciprofloxacin_NWTpredictions_MICdistByMethod.pdf
PDF	Ciprofloxacin_NWTpredictions.pdf
CSV	Ciprofloxacin_NWTpredictions.tsv
CSV	Ciprofloxacin_predictionsFromRules.tsv
PDF	Ciprofloxacin_SIRpredictions_byMethod.pdf
CSV	Ciprofloxacin_SIRpredictions_byMethod.tsv
PDF	Ciprofloxacin_SIRpredictions_bySource.pdf
CSV	Ciprofloxacin_SIRpredictions_bySource.tsv
PDF	Ciprofloxacin_SIRpredictions_diskDist.pdf
PDF	Ciprofloxacin_SIRpredictions_diskDistByMethod.pdf
PDF	Ciprofloxacin_SIRpredictions_MICdist.pdf
PDF	Ciprofloxacin_SIRpredictions_MICdistByMethod.pdf
PDF	Ciprofloxacin_SIRpredictions.pdf
CSV	Ciprofloxacin_SIRpredictions.tsv



## MIC distribution of input data, by platform coloured by rule-based prediction



# AMRrulemakeR package

<https://github.com/AMRverse/AMRrulemakeR>

> view(cip\_rules\$rules)

(or open file: Ciprofloxacin\_AMRrules.tsv)

A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T			
1	ruleID	organism	gene	nodeID	mutation	variation	type	gene context	drug	phenotype	clinical category	breakpoint	breakpoint st	breakpoint ct	ecoff	ecoff standar	PMID	evidence code	evidence grade		
2	ECO1001	s_Escherichia	aac(6')-lb-cr	aac(6')-lb-cr	-	Gene present	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	-		ECO:0001103	low	No solo data.	
3	ECO1002	s_Escherichia	gyrA	gyrA	Asp87Asn	Protein variation	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
4	ECO1003	s_Escherichia	gyrA	gyrA	Asp87Gly	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
5	ECO1004	s_Escherichia	gyrA	gyrA	Asp87Tyr	Protein variation	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
6	ECO1005	s_Escherichia	gyrA	gyrA	Gly81Cys	Protein variation	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	2168148		ECO:0001103	low	Limited samples.	
7	ECO1006	s_Escherichia	gyrA	gyrA	Ser83Ala	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
8	ECO1007	s_Escherichia	gyrA	gyrA	Ser83Leu	Protein variation	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	#####		ECO:0001103	moderate	Conflicting evidence.	
9	ECO1008	s_Escherichia	gyrA	gyrA	Ser83Val	Protein variation	acquired	Ciprofloxacin	nonwildtype	I	disk zone >=	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
10	ECO1009	s_Escherichia	marR	marR	Ser3Asn	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	-		ECO:0001103	moderate	-	
11	ECO1010	s_Escherichia	oqxB	oqxB	-	Inactivating	r	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	19528276		ECO:0001103	low	Limited samples.
12	ECO1011	s_Escherichia	parC	parC	Ala56Thr	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	12654733		ECO:0001103	low	Limited samples.	
13	ECO1012	s_Escherichia	parC	parC	Ser57Thr	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	#####		ECO:0001103	low	Limited samples.	
14	ECO1013	s_Escherichia	parC	parC	Ser80Ile	Protein variation	acquired	Ciprofloxacin	-	-	-	-	-	-	#####	-	-	-	-		
15	ECO1014	s_Escherichia	parE	parE	Asp475Glu	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
16	ECO1015	s_Escherichia	parE	parE	Ile355Thr	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
17	ECO1016	s_Escherichia	parE	parE	Ile529Leu	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	14506034		ECO:0001103	low	Limited samples.	
18	ECO1017	s_Escherichia	qnrA1	qnrA1	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	11943863		ECO:0001103	low	Limited samples.	
19	ECO1018	s_Escherichia	qnrB	qnrB	-	Gene presence	acquired	Ciprofloxacin	wildtype	S	disk zone >=	EUCAST 2025	Non-meningeal	disk zone >=	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
20	ECO1019	s_Escherichia	qnrB19	qnrB19	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	18519717		ECO:0001103	moderate	-	
21	ECO1020	s_Escherichia	qnrB2	qnrB2	-	Gene presence	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	disk zone >=	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
22	ECO1021	s_Escherichia	qnrB4	qnrB4	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	16870791		ECO:0001103	moderate	Conflicting evidence.	
23	ECO1022	s_Escherichia	qnrB6	qnrB6	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	19392890		ECO:0001103	low	Limited samples.	
24	ECO1023	s_Escherichia	qnrB7	qnrB7	-	Gene presence	acquired	Ciprofloxacin	-	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	-	-	17561500		ECO:0001103	low	Limited samples.	
25	ECO1024	s_Escherichia	qnrS1	qnrS1	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	15673773		ECO:0001103	moderate	Conflicting evidence.	
26	ECO1025	s_Escherichia	qnrS13	qnrS13	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	I	MIC > 0.25 &	EUCAST 2025	Non-meningeal	disk zone < 2'	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
27	ECO1026	s_Escherichia	qnrS2	qnrS2	-	Gene presence	acquired	Ciprofloxacin	nonwildtype	I	MIC > 0.25 &	EUCAST 2025	Non-meningeal	MIC > 0.064 r	EUCAST 2025	16804843		ECO:0001103	low	Limited samples.	
28	ECO1027	s_Escherichia	soxR	soxR	Gly121Asp	Protein variation	acquired	Ciprofloxacin	-	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	-	-			ECO:0001103	low	Limited samples.	
29	ECO1028	s_Escherichia	soxS	soxS	Ala12Ser	Protein variation	acquired	Ciprofloxacin	wildtype	S	MIC <= 0.25 r	EUCAST 2025	Non-meningeal	MIC <= 0.064	EUCAST 2025	-		ECO:0001103	low	Limited samples.	
30	ECO1029	s_Escherichia	acrR	acrR	Arg45Cys	Protein variation	acquired	Ciprofloxacin	-	S	-	-	-	-	-	-	ECO:0001103	low	Limited assay data		

# Curation of rules proposed using AMRrulemakeR

It is important that all rules are manually reviewed and curated by experts who have knowledge of the organism, the drug, the resistance mechanisms, and relevant literature.

While we have automated the process of proposing and grading rules based on quantitative data analysis, these rules needs to be checked carefully to make sure:

- They are supported by the quantitative data available, which are trustworthy and consistent with no obvious problems.
- They make biological sense, as far as you can tell.
- The PMID recorded is relevant to the specific organism, drug, and marker to which the rule applies.
- Additional evidence from the literature is recorded in the evidence code and PMID fields, and is considered appropriately in assigning the final evidence grade (e.g. some rules may be supported by very limited data and thus have a 'low' grade proposed, but be well supported by experimental data reported in the literature that justify a higher grade).

## Next steps:

- We are preparing a guide for running AMRrulemakeR, and a guide for curation of proposed rules
- Feb/March - meeting of subgroups leads (and nominated reps) to discuss use of AMRrulemakeR and subgroup curation of rules

# Agenda

- 1. Online resources:** where to find things
- 2. AMRrules Python package:** applying rules to interpret genotypes
- 3. AMRrulemakeR package:** defining rules from geno-pheno data
- 4. 2026 planning**
  - Manuscript plans
  - Meeting schedule
  - Events

## 2026 Planning - Manuscripts

1. AMRgen R package (Q1) Subgroup specific papers?
2. AMRrules (Q2)
  - *Concept* - aims, principles, working group model
  - *Specification* (v1.0) - format for how rules are specified, use of standard ontologies, new/modified ontologies/syntax
  - *Implementation* - how rules are applied to interpret AMRfp output (Python package v1.0)
3. Core gene rules (Q2)
  - Why interpreting core gene rules is important but challenging
  - Approach to defining rules
  - Summary of rules - comparison across pathogens, gaps in knowledge
4. Quantitative approach and acquired gene rules (Q4)

# AMRgen R package (Q1)

**Analysis group leads:** Matthijs Berends (Data & Tools group) & Kat Holt

***Contributors – code development:***

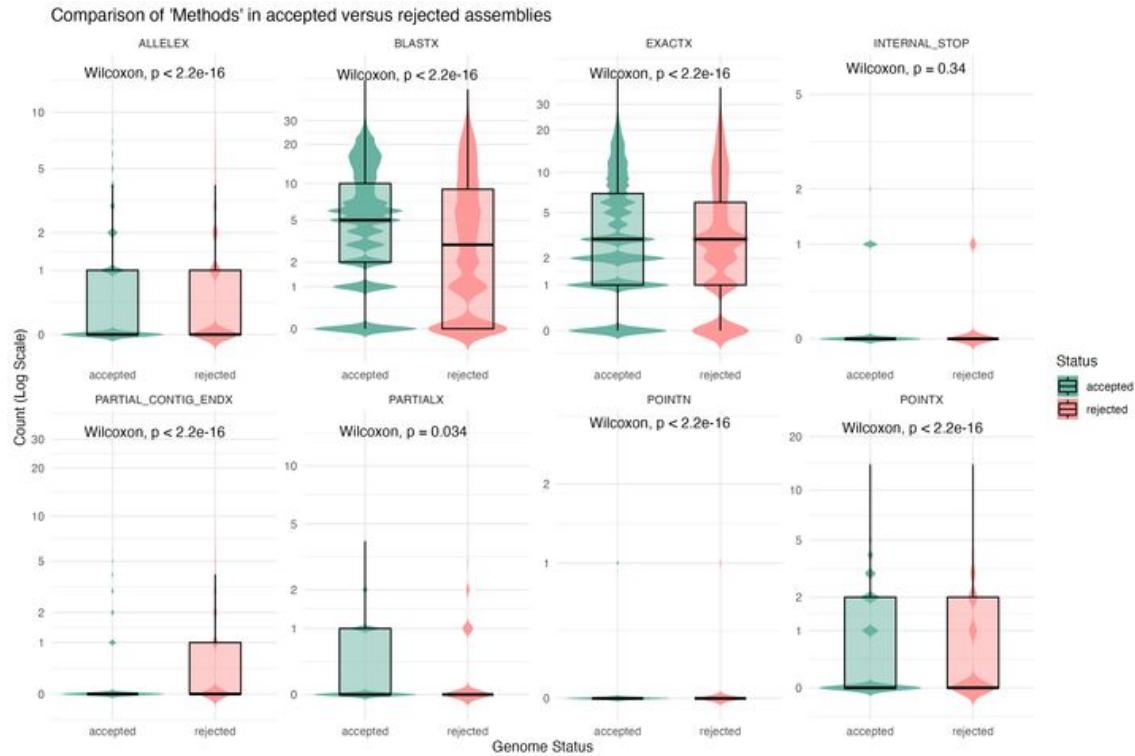
- all those who contributed (including via the Jan 2025 hackathon)
- please volunteer if you want to help add or refine functions

***Contributors – testing and examples:***

- please volunteer if you want to help develop example use cases to include in the paper (and in package documentation and vignettes)

# Qualibact paper

- We have assessed the impact of rejected assemblies on the results from AMRFinderPlus (still waiting for MLST results).
- Noticed the number of PARTIAL\_CONTIG\_ENDX was significantly higher in rejected assemblies.
- Waiting for subgroup assemblies to improve some of thresholds.



## 2026 Planning

1. **Fixed monthly meeting dates** - *Tuesdays 9am/5pm UK time*
  - Feb 24, Mar 17, Apr 28, May 19, Jun 23, Jul 21, Aug 18
2. **Who is attending Wellcome AMR?** (23-25 March, Hinxton, UK)
3. **ESGEM-AMR/Allthebacteria hackathon** (26-27 Cambridge, UK)
  - Sign up for hackathon (in person, Cambridge UK, 26-27 March):  
<https://forms.gle/a9vbFDhZh6veXbBFA>
4. **ESCMID global meetup?** (17-21 April, München, Germany)
5. **Collaboration with WHO AMR team on AMR Gene Catalog**

# Questions? / Any other business?

**ESGEM-AMR**



[amrrules.org](http://amrrules.org)