QMRA Assessment Toolkit

*User Guide and Technical Documentation*

**Developed by:** NIWA Earth Sciences New Zealand  
**Version:** 1.0  
**Date:** September 2025

# Table of Contents

1. Overview  
2. Installation  
3. Quick Start  
4. Command-Line Interface  
5. Python API  
6. Exposure Routes  
7. Pathogen Database  
8. Treatment Modeling  
9. Report Generation  
10. Examples  
11. Troubleshooting  
12. Best Practices

# 1. Overview

The QMRA Assessment Toolkit is a comprehensive Python-based solution for Quantitative Microbial Risk Assessment, developed by NIWA Earth Sciences New Zealand. This toolkit replaces @Risk Excel functionality with automated, reproducible workflows for regulatory compliance QMRA assessments.

## Key Features

* Comprehensive Pathogen Database - Validated dose-response models for key pathogens
* Multiple Exposure Routes - Primary contact, shellfish consumption, drinking water
* Dilution Modeling Integration - NIWA's key differentiator with engineer-provided LRVs
* Monte Carlo Simulation - Advanced uncertainty analysis replacing @Risk functionality
* Risk Characterization - Complete risk metrics calculation (infection, illness, DALYs)
* Automated Reporting - Generate regulatory compliance reports in Word format
* Command-Line Interface - Easy-to-use CLI for common workflows

## Benefits Over @Risk Excel

|  |  |  |
| --- | --- | --- |
| **Feature** | **@Risk** | **QMRA Toolkit** |
| Platform | Excel-dependent | Native Python |
| Security | Firewall conflicts | No external dependencies |
| Automation | Manual processes | Fully automated |
| Reproducibility | Limited | Complete version control |
| Cost | Commercial license | Open source |
| Integration | Limited | NIWA dilution modeling |

# 2. Installation

## Requirements

* Python 3.8 or higher
* NumPy and SciPy for numerical computations
* Pandas for data handling
* Matplotlib for plotting
* python-docx for report generation
* PyYAML for configuration files
* Click for command-line interface

## Setup Instructions

1. 1. Navigate to the toolkit directory:

cd qmra\_toolkit

1. 2. Install dependencies:

pip install -r requirements.txt

1. 3. Verify installation:

python tests/run\_all\_tests.py

1. 4. Test command-line interface:

python src/qmra\_toolkit.py --help

# 3. Quick Start

## Basic Risk Assessment Example

Run a complete QMRA assessment for recreational water exposure to norovirus:

python src/qmra\_toolkit.py assess \  
 --pathogen norovirus \  
 --exposure-route primary\_contact \  
 --concentration 10.0 \  
 --volume 50.0 \  
 --frequency 10 \  
 --population 10000 \  
 --report

This command will:

* Assess norovirus exposure from recreational water
* Use 10 organisms per 100mL water concentration
* Assume 50 mL water ingestion per swimming event
* Calculate risk for 10 swimming events per year
* Apply results to a population of 10,000
* Generate a comprehensive compliance report in Word format

## Example Output

QMRA Assessment Results: Norovirus  
============================================================  
  
Infection Probability:  
 Mean: 1.65e-01  
 Median: 1.65e-01  
 95th Percentile: 1.65e-01  
  
Annual Risk:  
 Mean: 8.36e-01  
 Median: 8.36e-01  
 95th Percentile: 8.36e-01  
 Expected cases per year: 8361  
  
Regulatory Compliance:  
 recreational\_water\_risk: FAIL  
 acceptable\_annual\_risk: FAIL

# 4. Command-Line Interface

The QMRA toolkit provides a comprehensive command-line interface for common workflows.

## Main Commands

### assess - Complete Risk Assessment

Performs a complete QMRA assessment for specified pathogen and exposure scenario.

* Required Parameters:
* --pathogen / -p: Pathogen name (norovirus, campylobacter, cryptosporidium)
* --exposure-route / -e: Exposure route (primary\_contact, shellfish\_consumption)
* --concentration / -c: Pathogen concentration (organisms per unit)
* Optional Parameters:
* --volume: Exposure volume (mL for water, grams for food)
* --frequency: Exposure frequency (events per year)
* --population: Population size for population risk calculations
* --iterations / -n: Number of Monte Carlo iterations (default: 10,000)
* --output / -o: Output file for results (JSON format)
* --report / -r: Generate comprehensive Word report

### list-pathogens - Available Pathogens

Lists all available pathogens in the database.

python src/qmra\_toolkit.py list-pathogens

### pathogen-info - Pathogen Details

Get detailed information about a specific pathogen.

python src/qmra\_toolkit.py pathogen-info --pathogen norovirus

# 5. Python API

For advanced users, the toolkit provides a comprehensive Python API for integration into custom workflows and applications.

## Basic API Usage Example

from pathogen\_database import PathogenDatabase  
from exposure\_assessment import create\_exposure\_assessment, ExposureRoute  
from risk\_characterization import RiskCharacterization  
  
# Initialize components  
pathogen\_db = PathogenDatabase()  
risk\_calc = RiskCharacterization(pathogen\_db)  
  
# Configure exposure  
exposure\_params = {  
 "water\_ingestion\_volume": 50.0, # mL per event  
 "exposure\_frequency": 10 # events per year  
}  
  
exposure\_model = create\_exposure\_assessment(  
 ExposureRoute.PRIMARY\_CONTACT,  
 exposure\_params  
)  
exposure\_model.set\_pathogen\_concentration(10.0)  
  
# Run assessment  
results = risk\_calc.run\_comprehensive\_assessment(  
 pathogen\_name="norovirus",  
 exposure\_assessment=exposure\_model,  
 population\_size=10000  
)  
  
# Print results  
for metric\_name, result in results.items():  
 print(f"{metric\_name}: {result.statistics['mean']:.2e}")

# 6. Exposure Routes

## Primary Contact (Recreational Water)

Models inadvertent water ingestion during swimming, surfing, or other water activities.

Required Parameters:

* water\_ingestion\_volume: Volume ingested per event (mL)
* exposure\_frequency: Number of exposure events per year

Typical Concentrations:

* Beach water: 1-100 organisms per 100 mL
* River water: 10-1000 organisms per 100 mL
* Treated wastewater discharge: 0.1-10 organisms per 100 mL

## Shellfish Consumption

Models pathogen exposure through shellfish consumption (oysters, mussels, clams).

Required Parameters:

* shellfish\_consumption: Mass consumed per serving (grams)
* consumption\_frequency: Number of servings per year

Typical Concentrations:

* Raw oysters: 100-10,000 organisms per 100g
* Cooked shellfish: 1-100 organisms per 100g

# 7. Pathogen Database

## Available Pathogens

The toolkit includes dose-response models for:

* Norovirus
* Model: Beta-Poisson (α=0.04, β=0.055)
* Source: Teunis et al. (2008)
* Illness-to-infection ratio: 0.7
* Campylobacter jejuni
* Model: Beta-Poisson (α=0.145, β=7.59)
* Source: Teunis et al. (2005)
* Illness-to-infection ratio: 0.3
* Cryptosporidium parvum
* Model: Exponential (r=0.0042)
* Source: Haas et al. (1996)
* Illness-to-infection ratio: 1.0

# 8. Treatment and Dilution Modeling

The toolkit includes sophisticated dilution modeling capabilities that integrate engineer-provided Log Reduction Values (LRVs), multi-barrier treatment trains, and uncertainty quantification - NIWA's key differentiator.

## Treatment Configuration

Create a YAML configuration file for treatment trains:

treatment\_barriers:  
 - name: "Primary Settling"  
 type: "physical"  
 lrv: 0.5  
 variability: 0.2  
  
 - name: "Activated Sludge"  
 type: "biological"  
 lrv: 2.0  
 variability: 0.5  
  
 - name: "UV Disinfection"  
 type: "uv\_disinfection"  
 lrv: 3.0  
 variability: 0.3

## Python API Usage

from dilution\_model import DilutionModel, TreatmentBarrier, TreatmentType  
  
# Create dilution model  
model = DilutionModel()  
  
# Add treatment barriers  
model.add\_treatment\_barrier(TreatmentBarrier(  
 name="UV Treatment",  
 treatment\_type=TreatmentType.UV,  
 log\_reduction\_value=3.0,  
 variability=0.2  
))  
  
# Apply to concentration  
initial\_conc = 1e6 # organisms/L  
results = model.apply\_complete\_scenario(initial\_conc)

# 9. Automated Report Generation

The toolkit automatically generates comprehensive regulatory compliance reports in professional Word format.

## Report Contents

1. Executive Summary with key findings
2. Introduction and Assessment Objectives
3. Methodology and Model Selection
4. Risk Assessment Results with statistical analysis
5. Uncertainty Analysis and Monte Carlo results
6. Regulatory Compliance Assessment
7. Conclusions and Recommendations
8. Technical Appendices with detailed data

## Generating Reports

Reports are automatically generated when using the --report flag:

python src/qmra\_toolkit.py assess \  
 --pathogen norovirus \  
 --exposure-route primary\_contact \  
 --concentration 10.0 \  
 --report

# 10. Advanced Examples

The toolkit includes advanced example scripts demonstrating complex analysis workflows.

## Multi-Pathogen Risk Comparison

Compare risk levels across different pathogens under identical conditions:

cd examples  
python pathogen\_comparison.py

This example demonstrates:

* Automatic model selection for different pathogens
* Risk ranking and comparative analysis
* Population-level risk calculations
* Regulatory compliance assessment across pathogens

## Treatment Scenario Comparison

Evaluate effectiveness of different treatment levels:

cd examples  
python scenario\_comparison.py

This example demonstrates:

* Treatment barrier modeling with log reduction values
* Risk reduction analysis across treatment scenarios
* Cost-benefit evaluation of treatment investments

# 11. Troubleshooting

## Common Issues

* **Import Errors:**
* Ensure all dependencies are installed: pip install -r requirements.txt
* **Pathogen Not Found:**
* Check available pathogens with: python src/qmra\_toolkit.py list-pathogens
* **Configuration Issues:**
* Verify YAML syntax in treatment configuration files
* **Memory Issues:**
* Reduce Monte Carlo iterations for large simulations using --iterations parameter
* **Report Generation Fails:**
* Ensure write permissions for output directory and python-docx is installed

## Getting Help

* Run the test suite: python tests/run\_all\_tests.py
* View CLI help: python src/qmra\_toolkit.py --help
* Check pathogen information: python src/qmra\_toolkit.py pathogen-info --pathogen [name]
* Review example scripts in the examples/ directory

# 12. Best Practices

## Model Selection

* Use Beta-Poisson models for most viral and bacterial pathogens
* Use Exponential models for highly infectious pathogens or conservative estimates
* Validate parameters against published literature
* Document model selection rationale in reports

## Uncertainty Analysis

* Use at least 10,000 iterations for robust uncertainty estimates
* Include parameter uncertainty in LRV values
* Document assumptions clearly in reports
* Consider sensitivity analysis for key parameters

## Quality Assurance

* Run validation tests regularly
* Compare results with previous assessments
* Review reports before submission
* Archive model inputs with results
* Use version control for project files

# Appendix: Technical Specifications

## System Requirements

* Operating System: Windows, macOS, or Linux
* Python: Version 3.8 or higher
* Memory: Minimum 4GB RAM (8GB recommended for large simulations)
* Disk Space: 100MB for toolkit + space for output reports
* Network: Internet connection for initial package installation

## File Structure

qmra\_toolkit/  
├── src/ # Core modules  
│ ├── pathogen\_database.py  
│ ├── dose\_response.py  
│ ├── exposure\_assessment.py  
│ ├── dilution\_model.py  
│ ├── monte\_carlo.py  
│ ├── risk\_characterization.py  
│ ├── report\_generator.py  
│ └── qmra\_toolkit.py # Main CLI  
├── data/ # Pathogen database  
├── tests/ # Test suite  
├── examples/ # Usage examples  
├── docs/ # Documentation  
└── requirements.txt # Dependencies

QMRA Assessment Toolkit v1.0  
© NIWA Earth Sciences New Zealand  
Strategic Investment Programme 2025-2026