Package ‘OAtools’

May 5, 2026

Title Analysis of OpenArray PCR Data

Version 1.0.0

Description Provides a suite of R functions to analyze gene expression experiments on the OpenArray real-time PCR platform. OAtools fits logistic regressions to fluorescence curves to distinguish between real amplification and false positives. OAtools supports data import, analysis, and visualization through plots and a dynamic HTML report.

License GPL (>= 3)

LazyData FALSE

Depends R (>= 4.6)

Imports basilisk (>= 1.20.0), Biobase (>= 2.70.0), dplyr (>= 1.1.4), DT (>= 0.34.0), ggplot2 (>= 3.5.2), janitor (>= 2.2.1), methods (>= 4.5.2), plotly (>= 4.11.0), purrr (>= 1.2.0), ReadqPCR (>= 1.56.0), readxl (>= 1.4.5), reticulate (>= 1.43.0), rlang (>= 1.1.6), rmarkdown (>= 2.29), S4Vectors (>= 0.48.0), shiny (>= 1.13.0), SummarizedExperiment (>= 1.40.0), tibble (>= 3.3.0), tidyr (>= 1.3.1), writexl (>= 1.5.4)

Suggests testthat (>= 3.0.0), knitr, kableExtra (>= 1.4.0), NormqPCR (>= 1.56.0), BiocManager (>= 1.30.27)

biocViews qPCR, GeneExpression, DataImport, Regression

BugReports `https://github.com/uwvirology-ngs/OAtools/issues`

URL `https://github.com/uwvirology-ngs/OAtools`

BiocType Software

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Encoding UTF-8

VignetteBuilder knitr

StagedInstall no

Config/Bioconductor/UnsupportedPlatforms windows-x86

git_url https://git.bioconductor.org/packages/OAtools

git_branch RELEASE_3_23

git_last_commit 1cc067d

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-05-04

Author Aidan Shea [aut, cre] (ORCID: <`https://orcid.org/0009-0001-9256-7439`>)

Maintainer Aidan Shea <`aidantshea@gmail.com`>

Contents

`.buildServer` *Web Server*

Description

Helper method to buildApp() responsible for construction of the Shiny web server

Usage

```
.buildServer(input, output, session)
```

Arguments

`input`	reactive values from UI widgets
`output`	render functions for results
`session`	environment for user session

Value

A Shiny web server

`.buildUI` *User Interface*

Description

Helper method to buildApp() responsible for construction of the graphical user interface

Usage

```
.buildUI()
```

Value

A Shiny UI

`.runFitCurve` *Run the fit_curve() function*

Description

A thin R wrapper for the python3 function fit_curve(), which attempts to optimize 5-parameter logistic regressions to PCR fluorescence curves.

Usage

```
.runFitCurve(pcr_data, linear_threshold)
```

Arguments

`pcr_data`	A `data.frame` with the following required columns: **cycle** PCR cycle number **fluo** observed fluorescence
`linear_threshold`	numeric value specifying the minimum overall change-in-fluorescence over the PCR reaction required for the optimizer to attempt fitting a logistic model

Value

The model as a nested list

`.serverDataImport` *Data Import Server*

Description

Data Import Server

Usage

```
.serverDataImport(id)
```

Arguments

`id`	identifier unique to `Data Import` page

Value

The `Data Import` server component

`.serverFitCurves` *Curve Fitting Server*

Description

Curve Fitting Server

Usage

```
.serverFitCurves(id, se)
```

Arguments

`id`	identifier unique to `Fit Curves` page
`se`	SummarizedExperiment with OpenArray assay data

Value

The `Fit Curves` server component

`.serverGraphics` *Graphics Server*

Description

Graphics Server

Usage

```
.serverGraphics(id, se)
```

Arguments

`id`	identifier unique to `Graphics` page
`se`	SummarizedExperiment with OpenArray assay data

Value

The `Graphics` server component

`.serverReporting` *Reporting Server*

Description

Reporting Server

Usage

```
.serverReporting(id, se_fit)
```

Arguments

`id`	identifier unique to `Reporting` page
`se_fit`	SummarizedExperiment with OpenArray assay data

Value

The `Reporting` server component

`.uiDataImport` *Data Import UI*

Description

Data Import UI

Usage

```
.uiDataImport(id)
```

Arguments

`id`	identifier unique to `Data Import` page

Value

The `Data Import` page

`.uiFitCurves` *Curve Fitting UI*

Description

Curve Fitting UI

Usage

```
.uiFitCurves(id)
```

Arguments

`id`	identifier unique to `Fit Curves` page

Value

The `Fit Curves` page

`.uiGraphics` *Graphics UI*

Description

Graphics UI

Usage

```
.uiGraphics(id)
```

Arguments

`id`	identifier unique to `Graphics` page

Value

The `Graphics` page

`.uiReporting` *Reporting UI*

Description

Reporting UI

Usage

```
.uiReporting(id)
```

Arguments

`id`	identifier unique to `Reporting` page

Value

The `Reporting` page

`buildApp` *A Shiny App for OAtools*

Description

Builds a Shiny application for interactively running OAtools

Usage

```
buildApp()
```

Details

Temporarily changes the maximum file size for upload to 30MB, restoring the original setting upon application exit.

Value

A Shiny web application

Examples

```
app <- buildApp()
```

`computeModels` *Optimize 5PL models to OpenArray PCR data and save as metadata*

Description

Computes 5-parameter logistic models optimized to PCR data contained in the specified assay matrix, either `fluo_normalized` or `fluo_reporter`. The former refers to the base-line adjusted, normalized fluorescence from the `Amplification Data` tab of the original Excel output. The latter refers to the multicomponent fluorescence from the `Multicomponent Data` tab.

The expected input for this function is a SummarizedExperiment object containing OpenArray PCR data, which can be generated by calling the `excelToSE()` function of the package on the raw Excel output from QuantStudio software.

Usage

```
computeModels(se, assay_name, linear_threshold = 400)
```

Arguments

`se`	OpenArray PCR data as a SummarizedExperiment object
`assay_name`	character value specifying the assay matrix from which to load PCR data for curve-fitting, either `fluo_normalized` or `fluo_reporter`
`linear_threshold`	numeric value specifying the minimum overall change-in-fluorescence over the PCR reaction required for the optimizer to attempt fitting a logistic model

Details

Under the hood, this function invokes .runFitCurve(), a thin wrapper which calls python3 code to fit models with scipy.optimize, on each PCR reaction of the OpenArray plate separately. The computed models are added to the SummarizedExperiment container in metadata, named as `assay_name` + `_models`. For example, `fluo_normalized_models` would be created in the experiment metadata if `computeModels(assay_name = "fluo_normalized")` is called.

Value

OpenArray PCR data as a SummarizedExperiment object with information from the computed model stored as metadata.

Examples

```
path <- system.file(
    "extdata",
    "oa_gene_expression_1.xlsx",
    package = "OAtools"
)
```

```
se <- excelToSE(excel_path = path) |>
    computeModels(assay_name = "fluo_normalized") |>
    computeModels(assay_name = "fluo_reporter")
```

`determinePCRResults` *Determine PCR results from fit curves*

Description

Assigns positive or negative PCR results to each reaction depending on the equation of the model optimized to observed multicomponent fluorescence values. The reporter dye fluorescence without normalization is used to distinguish between real amplification and false positives.

Usage

```
determinePCRResults(se, key_path)
```

Arguments

`se`	a SummarizedExperiment object containing OpenArray qPCR Data
`key_path`	file path to the target-threshold key

Details

The key is an Excel file storing values used by `determinePCRResults()` to categorize PCR curves as `positive` or `negative`. In the key, each gene is associated with threshold values for Crt, change-in-fluorescence, and slope at the reaction midpoint.

For more specifics, input `?target_threshold_key` into the R console.

Value

a SummarizedExperiment

Examples

```
data(example_se)

key_path = system.file(
    "extdata",
    "target_threshold_key.xlsx",
    package = "OAtools"
)

se <- example_se |>
    determinePCRResults(key_path = key_path)
```

`example_se` *Example OpenArray Gene Expression Data Contained in a SummarizedExperiment*

Description

A sample of OpenArray gene expression data from respiratory tract microbiota profiling experiments conducted at the UW Virology research lab on human nasal swabs. This file (.rda) stores the PCR data in a SummarizedExperiment container and is intended for use with package examples and unit testing.

Format

a SummarizedExperiment object with the following:

colData information associated with each PCR well

rowData cycle numbers

assays matrices of fluorescence values by PCR well and cycle

Details

Load this object into the environment with: `data(example_se)`

Source

This object contains PCR data imported from the initial Excel QuantStudio output. Logistic and linear models were fit to the PCR data and stored as metadata.

The object can be reproduced from the package example data by running the following commands in the R console:

```
path <- system.file(
    "extdata",
    "oa_gene_expression_1.xlsx",
    package = "OAtools"
)

se <- excelToSE(excel_path = path) |>
    computeModels(assay_name = "fluo_normalized") |>
    computeModels(assay_name = "fluo_reporter")
```

`excelToSE` *Convert OpenArray data from Excel to a Summarized Experiment*

Description

Transforms raw gene expression run data exported from the OpenArray QuantStudio 12K Flex Software from .xlsx format into an instance of the SummarizedExperiment class from Bioconductor.

Usage

```
excelToSE(excel_path, header_rows = 17, skip = 19)
```

Arguments

`excel_path`	file path to the Excel document containing the PCR data
`header_rows`	number of rows of run metadata to read in from the header
`skip`	number of rows to skip when reading fluorescence data or results

Value

OpenArray PCR data as a SummarizedExperiment object

Examples

```
path = system.file(
    "extdata",
    "oa_gene_expression_1.xlsx",
    package = "OAtools"
)

se <- excelToSE(excel_path = path)
```

`generateReport` *Generate a PCR Report*

Description

Knits an HTML report summarizing the OpenArray experiment and saves to the specified directory.

Usage

```
generateReport(se, path = tempdir(), model_results = FALSE)
```

Arguments

`se`	a SummarizedExperiment containing OpenArray qPCR data
`path`	intended outfile path, defaults to a temporary directory
`model_results`	boolean value indicating whether to include the `results` column, which is created when deriving results using the curve-fitting method

Value

An HTML Report summarizing the OpenArray experiment

Examples

```
data(example_se)
generateReport(se = example_se)
```

`oa_gene_expression_1` *Example Raw OpenArray Gene Expression Data*

Description

The first of two Excel files storing run data from separate OpenArray gene expression experiments. The context behind the experiment was respiratory tract microbiota profiling on human nasopharyngeal swabs from patients experiencing respiratory syndromes.

Format

An Excel file (`.xlsx`) with three tabs:

Amplification Data normalized fluorescence by cycle number

Multicomponent Data spectral contribution of the reporter dye by cycle number

Results PCR results, PCR well metadata, and QC metrics

Details

Access this file with: `system.file("extdata", "oa_gene_expression_1.xlsx", package = "OAtools")`

Source

This file was exported from QuantStudio 12K Flex Software after a gene expression run, then filtered down to include 12 samples and 4 genes for file size concerns.

`oa_gene_expression_2` *Example Raw OpenArray Gene Expression Data*

Description

The second of two Excel files storing run data from separate OpenArray gene expression experiments. The context behind the experiment was respiratory tract microbiota profiling on human nasopharyngeal swabs on patients experiencing respiratory syndromes.

Format

An Excel file (`.xlsx`) with three tabs:

Amplification Data normalized fluorescence by cycle number

Multicomponent Data spectral contribution of the reporter dye by cycle number

Results PCR results, sample metadata, and QC metrics

Details

Access this file with: `system.file("extdata", "oa_gene_expression_2.xlsx", package = "OAtools")`

Source

This file was exported from QuantStudio 12K Flex Software after a gene expression run, then filtered down to include 12 samples and 4 genes for file size concerns.

`plotCrt` *Plot Relative Cycle Threshold Values by Gene*

Description

Generates a box and whisker plot visualizing the distribution of Crt values measured on an OpenArray plate by Gene.

Usage

```
plotCrt(se)
```

Arguments

`se`	a SummarizedExperiment object containing OpenArray qPCR data

Value

a ggplot2 figure

Examples

```
data(example_se)

plotCrt(example_se)
```

`plotModel` *Plot Fluorescence Values Predicted by Model*

Description

Juxtaposes the fluorescence values predicted by the model optimized to the measured fluorescence vs. cycle data for a particular well.

Usage

```
plotModel(
  se,
  well_id,
  assay_name,
  include_mdpt_tangent = FALSE,
  include_coldata_annotation = FALSE
)
```

Arguments

`se`	a SummarizedExperiment object containing OpenArray qPCR data
`well_id`	a character representing the name of the well to plot as listed in the assay matrix
`assay_name`	name for the assay matrix from which to pull observed fluorescence values, either `fluo_reporter` or `fluo_normalized`
`include_mdpt_tangent`	boolean determines whether to annotate the midpoint of the reaction and draw a tangent line to the model curve at that point
`include_coldata_annotation`	boolean determines whether to annotate the coldata onto the top left of the plot.

Value

a ggplot2 figure

Examples

```
data(example_se)

plotModel(
    example_se,
    well_id = "well_2665",
    assay_name = "fluo_reporter",
```

```
    include_mdpt_tangent = TRUE,
    include_coldata_annotation = TRUE
)
```

`plotOverview` *Plot Amplification Status by Sample and Gene*

Description

Generates an overview graphic summarizing qPCR results for each combination of sample and gene on an OpenArray experiment.

Usage

```
plotOverview(se)
```

Arguments

`se`	a SummarizedExperiment object containing OpenArray qPCR data

Value

a ggplot2 figure

Examples

```
data(example_se)

plotOverview(example_se)
```

`plotQC` *Plot a 3D Quality Control Graphic from a SummarizedExperiment*

Description

Generates a 3-dimensional quality control plot comparing the amplification status to the crt, Cq conf, and amplification score metrics output by QuantStudio 12K Flex Software.

Usage

```
plotQC(se)
```

Arguments

`se`	a SummarizedExperiment object containing OpenArray qPCR data

Value

a plotly figure

Examples

```
data(example_se)
plotQC(example_se)
```

`seToQPCRBatch` *Convert to qPCRBatch Object*

Description

Transforms OpenArray run data contained within the SummarizedExperiment container into a qPCRBatch object. This conversion allows for convenient gene expression analyses with the NormqPCR package.

Usage

```
seToQPCRBatch(se)
```

Arguments

`se`	A SummarizedExperiment object with OpenArray run data

Value

a qPCRBatch object

Examples

```
path = system.file(
    "extdata",
    "oa_gene_expression_1.xlsx",
    package = "OAtools"
)

se <- excelToSE(excel_path = path)

qpcr <- seToQPCRBatch(se)
```

`target_threshold_key` *Target Threshold Key*

Description

This Excel file (.xlsx) contains a table associating each assay target contained in the example gene expression run data with thresholds values. This key is optionally used to aid in concurrently interpreting data including numerous targets with dissimilar behaviors.

Format

An Excel sheet (`.xlsx`) with four columns:

`target`	the target assay ID
`slope_threshold`	minimum acceptable slope in the exponential phase for a positive result
`delta_threshold`	minimum acceptable overall change in fluorescence for a positive result
`crt_threshold`	maximum acceptable crt for a positive result

Details

The example key is stored in the `inst/extdata/` directory of the package. Access this file with:
`system.file("extdata", "target_threshold_key.xlsx", package = "OAtools")`.

Source

Written manually for use with resulting functions.

Index