Package 'MDMAPR'

July 21, 2025

Type Package

Title Molecular Detection Mapping and Analysis Platform

Version 0.2.3

Author Alka Benawra <alkabenawra@rogers.com>

Maintainer Alka Benawra <alkabenawra@rogers.com>

Description Runs a Shiny web application that merges raw 'qPCR' fluorescence data with related metadata to visualize species presence/absence detection patterns and assess data quality. The application calculates threshold values from raw fluorescence data using a method based on the second derivative method, Luu-The et al (2005) <doi:10.2144/05382RR05>, and utilizes the 'chipPCR' package by Rödiger, Burdukiewicz, & Schierack (2015) <doi:10.1093/bioinformatics/btv205> to calculate Cq values. The application has the ability to connect to a custom developed MySQL database to populate the applications interface. The application allows users to interact with visualizations such as a dynamic map, amplification curves and standard curves, that allow for zooming and/or filtering. It also enables the generation of customized exportable reports based on filtered mapping data.

License GPL-3

Encoding UTF-8

Imports RMySQL, shinydashboard, DBI, DT, leaflet, leaflet.extras, shinyjs, ggplot2, dplyr, readxl, plotly, reactable, writexl, xfun, berryFunctions, shinyWidgets, shiny, htmltools, methods, utils, stats, bslib, htmlwidgets

RoxygenNote 7.1.1

URL https://github.com/HannerLab/MDMAPR

BugReports https://github.com/HannerLab/MDMAPR/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-23 04:20:07 UTC

Contents

addThresholdCq 2
dbInstance 3
dbVariables 3
formatRawFluorescenceFile 4
launchApp 5

Index **6**

`addThresholdCq` *Add systemCalculatedThresholdValue, systemCalculatedCqValue, and userProvidedCqValue to results_Table or standardCurveResults_Table files.*

Description

Function adds systemCalculatedThresholdValue and systemCalculatedCqValue to results_Table or standardCurveResults_Table files. User can also add userProvidedCqValue to file by setting calculateUserProvidedCq parameter to "Yes". If user wants system to calculate userProvidedCqValue they must provide userProvidedThresholdValue's.

Usage

```
addThresholdCq(file, calculateUserProvidedCq)
```

Arguments

`file`	results_Table or standardCurveResults_Table file that is read in.
`calculateUserProvidedCq`	defines if user wants UserProvidedCq value to be calculated and added to file. Update variable to "Yes" to have value added or "No" if not.

Value

Writes updated CSV file to local directory.

`dbInstance` *Indicates whether MDMAPR is connected to a database instance or not.*

Description

Function to indicate if MDMAPR will be run with a database instance. If MDMAPR is to be run with a database connection, update dbInstance to "Yes" and if not, update dbInstance to "No". If the application is run with a database connection update the dbVariables function with the correct database connection details before launching app.

Usage

```
dbInstance(x)
```

Arguments

x	defines if there is a database connection or not. Must be either updated to "Yes" or "No".

Details

dbInstance

Value

No return value.

Examples

```
dbInstance("Yes")
dbInstance("No")
```

`dbVariables` *Information needed to connect to a local or remote MDMAPR MySQL database instance.*

Description

Function stores username, password, database name, and host name for database instance to connect to. Please ensure dbInstance() is set to "Yes" if you are running the application with a database connection.

Usage

```
dbVariables(user, password, dbname, host)
```

Arguments

user	defines username for database connection.
password	defines password for database connection.
dbname	defines name of database connection.
host	defines host name for database connection.

Details

dbVariables

Value

No return value.

Examples

```
dbVariables(user = "root",
password = "Test23!",
dbname = 'MDMap_2.0',
host = "127.0.0.1")
```

---

formatRawFluorescenceFile

Format raw qPCR fluorescence file data for results_Table or standardCurveResults_Table.

Description

Function formats raw qPCR fluoresence file data from MIC, StepOnePlus or Biomeme two3/Franklin machines into table that includes well location names and fluorescence data for each reaction cycle. The table is written to the local machine directory as a CSV file. The formatted data can be copied and pasted into the results_Table or standardCurveResults_Table, which are used in the MDMAPR 2.0 MySQL database.

Usage

```
formatRawFluorescenceFile(rawFluorescenceFile, platform, outputFileName)
```

Arguments

rawFluorescenceFile	defines raw qPCR fluorescence file to read in.
platform	defines platform fluorescence file was generated from.
outputFileName	defines output name for CSV file that data will be written to.

Value

Writes CSV file to local directory.

launchApp *Launch the MDMAPR app.*

Description

This function runs the MDMAPR Shiny web application. Please update the dbInstance function first to specify if you are running the application with or without a database connection. If you are running the application with a database, then after updating the dbInstance function, run the dbVariables function to input the variables needed to establish a connection to your MDMAPR MySQL database instance. Then run launchApp() to launch app.

Usage

```
launchApp()
```

Value

shiny application object

Index

addThresholdCq, 2

dbInstance, 3
dbVariables, 3

formatRawFluorescenceFile, 4

launchApp, 5