Package 'qfa'

February 5, 2015

Version 0.0-33

Date 2015-01-21

Title Tools for Quantitative Fitness Analysis (QFA) of Arrayed Microbial Cultures Growing on Solid Agar Surfaces

Author Conor Lawless <conor.lawless@ncl.ac.uk>, with contributions from Alexander Young <alextisyoung@gmail.com> and Darren Wilkinson <d.j.wilkinson@ncl.ac.uk>

Maintainer Conor Lawless <conor.lawless@ncl.ac.uk>

Depends R (>= 2.10.1), sp, DEoptim, parallel

Imports knitr

Enhances data.table

Description Quantitative Fitness Analysis (QFA) is a complementary series of experimental and computational methods for estimating the fitness of thousands of microbial cultures in parallel. QFA is suitable for focussed, high-quality studies of the effect of genetic mutations or drug interventions on growth in model microbial organisms such as brewer's yeast. Culture growth is observed by time-lapse photography of solid agar plates inoculated with cultures in rectangular arrays. Growth curves are constructed by analysing image series using Colonyzer image analysis software (http://research.ncl.ac.uk/colonyzer) which converts images to arrays of cell density estimates. This R package is for a) fitting the generalised logistic model to potentially thousands of parallel growth curves, b) using inferred parameter values to calculate fitnesses for each culture and c) comparing fitnesses between QFA experiments with different genetic backgrounds or treatments to deduce interaction strengths. This package facilitates quantifying the fitness of thousands of independent microbial strains and tracking them throughout growth curve experiments. With appropriately designed experiments, qfa can also estimate genetic interaction strengths and produce epistasis plots.

License Artistic-2.0

VignetteBuilder knitr, utils

URL http://qfa.r-forge.r-project.org/

BugReports conor.lawless@ncl.ac.uk

R topics documented:

`colonyzer.read` *Reads raw cell density timecourse data from Colonyzer output files*

Description

Reads in and binds together all of the Colonyzer output files in a directory, so they are checked and ready for bayesian or likelihood inference. Colonyzer is an open source image analysis tool for quantifying cell densities on agar plates: http://research.ncl.ac.uk/colonyzer/

Usage

```
colonyzer.read(path=".",files=c(),experiment="ExptDescription.txt",
ORF2gene="",libraries="LibraryDescriptions.csv",screenID="")
```

Arguments

`path` The path to the folder containing the Colonyzer .dat files to be read: working directory by default.

`files` Character vector giving locations of Colonyzer .dat files to be read - overrides path

`experiment` Name of text file describing the inoculation times, library and plate number for unique plates. Taken relative to path if specified. File must be a tab-delimited text file with no header containing the following columns:

- Image.Name - Full name at image capture (includes barcode and date-time) of image from which data are derived
- Row - Row number (counting from top of image) of culture in rectangular gridded array
- Col - Column number (counting from left of image) of culture in rectangular gridded array
- X.Offset - x-coordinate of top left corner of rectangular tile bounding culture (number of pixels from left of image)
- Y.Offset - y-coordinate of top left corner of rectangular tile bounding culture (number of pixels from top of image)
- Area - Culture area (pixels)
- Trimmed - Integrated Optical Density, sum of pixel intensities within culture area
- Threshold - Global pixel intensity threshold used for image segmentation (after lighting correction)
- Intensity - Total pixel intensity for square tile containing culture
- Edge Pixels - Number of pixels classified as culture on edge of square tile
- Colony.Color.R - Culture red channel intensity
- Colony.Color.G - Culture green channel intensity
- Colony.Color.B - Culture blue channel intensity
- Background.Color.R - Background red channel intensity (for current tile)
- Background.Color.G - Background green channel intensity (for current tile)
- Background.Color.B - Background blue channel intensity (for current tile)
- Edge.length - Number of culture pixels classified as being microcolony edge pixels (useful for classifying contaminants in cultures grown from dilute inoculum)
- Tile.Dimensions.X - Culture tile width (pixels)
- Tile.Dimensions.Y - Culture tile height (pixels)
- Growth - Default measure of cell density (direct copy of one of Trimmed, Threshold or Intensity)
- Barcode - Unique plate identifier
- Date.Time - Timestamp of image capture (extracted from image filename)
- Inoc.Time - User specified date and time of inoculation (specified in Expt-Description.txt file)
- Treatments - Conditions applied externally to plates (e.g. temperature(s) at which cultures were grown, UV irradiation applied, etc.)
- Medium - Nutrients/drugs in plate agar
- Screen.Name - Name of screen (identifies biological repeats, and experiment)

- RepQuad - Integer identifying which of the quadrants of a 1536 plate were used to inoculate the current 384 plate (set equal to 1 for all cultures for 1536 format for example)
- MasterPlate Number - Library plate identifier
- Timeseries order - Sequential photograph number
- Library.Name - Name of library, specifying particular culture location
- ORF - Systematic, unique identifier for genotype in this position in arrayed library
- Gene - Standard gene name for genotype in this position in arrayed library. Note that this can be set equal to ORF for example
- ScreenID - Unique identifier for this QFA screen
- Client - Client for whom screen was carried out
- ExptDate - A representative/approximate date for the experiment (note that genome-wide QFA screens typically take weeks to complete)
- User - Person who actually carried out screen
- PI - Principal investigator leading project that screen is part of
- Condition - The most important defining characteristic of screen, as specified by user (e.g. the temperature screen was carried out at if screen is part of multi-temperature set of screens, or the query mutation if part of a set of screens comparing query mutations, or the drugs present in the medium if part of a set of drug screens)
- Inoc - Qualitative identifier of inoculation type (e.g. "DIL" for dilute inoculum, "CONC" for concentrated). Used to distinguish between experiments carried out with different methods of inoculation.
- Expt.Time - Time (days) since user-specified inoculation date (Inoc.Time) that current image was captured

`ORF2gene` Path to a tab-delimited text file containing two columns (with no headers) associating unique, systematic strain identifiers (e.g. yeast ORF Y-numbers) with human readable gene names (e.g. standard names from SGD).

`libraries` Tab-delimited text file describing the array strains present in each row-column coordinate of each plate in a series of rectangular arrayed libraries. Header row format is: "Library ORF Plate Row Column Notes". Columns are:

- Library - Library identifier (e.g. SDLV1)
- ORF - Systematic strain identifier
- Plate - Plate number
- Row - Row number
- Column - Column number
- Notes - Optional strain notes (e.g. is strain especially sick or missing?)

`screenID` Unique experiment identifier (e.g. QFA00001)

Value

An R data.frame where each row corresponds to a single observation on a single colony, with the value of the growth measurement in 'Growth', and the date and time of the measurement in 'Date.Time'. Other information about the observation is stored in the other columns. Several columns returned are direct copies of Colonyzer output and mapped as follows:

- Image.Name - Image Name
- Row - Spot Row

- Col - Spot Column
- X.Offset - X Offset
- Y.Offset - Y Offset
- Area - Area
- Trimmed - Trimmed Area
- Threshold - Threshold
- Intensity - Intensity
- Edge.Pixels - Edge Pixels
- Colony.Color.R - Colony Color R
- Colony.Color.G - Colony Color G
- Colony.Color.B - Colony Color B
- Background.Color.R - Background Color R
- Background.Color.G - Background Color G
- Background.Color.B - Background Color B
- Edge.length - Edge length
- Tile.Dimensions.X - Tile Dimensions X
- Tile.Dimensions.Y - Tile Dimensions Y

Extra columns are automatically added as follows. Some of this information is derived from auxiliary files passed to the function such as the experimental description file, the orf-gene dictionary and the library description file:

- Growth - A cell density surrogate built from trimmed Area normalised by tile area and maximum achievable pixel intensity: Trimmed/(Tile.Dimensions.X*Tile.Dimensions.Y*255)
- Barcode - Plate identifier, essentially image name with date time and file extension stripped
- Date.Time - Date time of image capture in YYYY-MM-DD_hh-mm-ss format
- Inoc.Time - Date time that plate was inoculated. If plate is grown at a high temperature, date time at which plate was moved into high temperature incubator. The assumption in this case being that negligible growth occurred before plate temperature was shifted the the target temperature.
- Treatments - Treatments applied to plate (e.g. temperature)
- Medium - Medium contained in agar (e.g. nutrients or drugs added to agar)
- Screen.Name - Unique identifier for experiment (usually identifies repeat number also if multiple repeats carried out).
- RepQuad - Identifier for experiments scaling down from 1536 format plates to 384, indicating which quadrant on the original 1536 source plate the current 384 format plate belongs to.
- MasterPlate.Number - Identifies which plate in the source library (as described in the library description file) corresponds to the current plate
- Timeseries.order - Ordinal describing which photograph captured
- Library.Name - Identifies which of the libraries identified in the library description file was used to construct this plate
- ORF - Unique systematic identifier for the genotype of the strain at this location (e.g. yeast Y-number), as defined by library description file
- Gene - Standard, human readable genotype identifier for the strain at this location, as defined by the ORF-Gene dictionary

- Background - Tag identifying experiment, typically used to construct file names and axes titles in plots
- Expt.Time - Number of days passed between inoculation (start of experiment) and current time

Finally, as well as returning the object above, this function prints a small report to screen, summarising the data returned. This includes number of unique barcodes read, number of photos read, number of genotypes in experiment, number of unique culture observations made, a list of treatments applied, a list of media used, a list of unique screen names (e.g. replicates carried out), the plate dimensions (e.g. 1536, 384 or 96 format) and a list of unique inoculation dates.

`correlationReport` *Correlation Report*

Description

Generates all possible fitness correlation plots, on a plate-by-plate basis, comparing all possible combinations of replicates for a given medium, treatment and plate number. Useful tool for searching for incorrect plate orientation, or misplaced/mislabelled plates. Can also give clues about plates with incorrect medium.

Usage

```
correlationReport(scrnms,dataframe,outputfile,aw=4,ah=4,fitmax=185)
```

Arguments

`dataframe`	Dataframe containing fitnesses to be summarised. Typically output from qfa.fit function. Must add a "fit" column before passing to this function.
`outputfile`	Output file name.
`scrnms`	Screen names to test for correlation problems
`fitmax`	Upper limit for both x and y axes in correlation plots. Maximum observable fitness for this combination of experiments and fitness definitions.
`aw`	Number of horizontal panels per page in output report.
`ah`	Number of vertical panels per page in output report.

Value

Generates a multi-page .pdf report demonstrating two-way between-replicate correlation for all possible pairs of plates and for all media, treatments and replicates in a QFA experiment.

data.fit — *Fitting generalised logistic model to growth data by least squares (using optim function)*

Description

Uses the optim function to carry out least squares fit of logistic model to growth data.

Usage

```
data.fit(tim,growth,inocguess,xybounds,inits=list(),logTransform=FALSE,verbose=FALSE)
```

Arguments

tim	Vector of culture size observation times
growth	Vector of observed culture sizes
inocguess	Inocluation density estimate (same physical units as growth vector)
xybounds	List of upper and lower bounds for model parameters. NEED TO ADD EXAMPLE OF CONTENTS.
inits	List of initial guesses for each parameter. If empty list, generate initial guesses from mean of bounds
logTransform	Boolean specifying whether we shoudl carry out model fit on log scale.
verbose	Boolean specifying level of reporting during model fitting

Value

A named vector of best parameter estimates along with the objective function (square distance between data and model) at the solution.

dtl — *Culture Doubling Time for Generalised Logistic Function (as a function of time t)*

Description

Calculates the Generalised Logistic Model (http://en.wikipedia.org/wiki/Generalised_logistic_function) version of culture doubling time as a function of time t.

Usage

```
dtl(K,r,g,v,t)
```

Arguments

K	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
r	Culture growth rate parameter (per day).
g	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
v	Shape parameter. Recover logistic model with v = 0.
t	Time since inoculation (d).

---

fitnessReport *Fitness Report*

Description

Summarises mean and median fitnesses for all orfs in qfa.fit object, and writes them to file with an appropriate header.

Arguments

treatment	Fitnesses are filtered by this value in the qfa.fit treatment column before summaries are generated.
outputfile	Output file name.
dataframe	Dataframe containing fitnesses to be summarised. Typically output from qfa.fit function. Must add a "fit" column before passing to this function.

Value

Writes summary fitnesses to file with header containing some metadata.

---

getDeadLocations *Find dead cultures in SGA plates (1536 format), and report their location in spotted plates (384 format).*

Description

Given Colonyzer quantifications of SGA plates in 1536 format, locate dead cultures and report the 384-format locations of those cultures for stripping.

Usage

```
getDeadLocations(SGAFile,SGAExpt,CutoffFrac=0.0025)
```

Arguments

SGAFile	The Colonyzer output for the final SGA plates. These are 1536 format plates, and the Colonyzer file should be a .dat tab delimited text file.
SGAExpt	Experiment description file for the SGA, linking barcode with plate number.
CutoffFrac	Optional argument for specifying the minimum value for Growth (normalised IOD) corresponding to detection of cells on 1536 plate.

Value

A data frame containing columns ROW384, COLUMN384, REP384 (repeat or quadrant identifier) and PLATE which will be useful for stripping.

`Glogist` *Generalised Logistic growth curve model*

Description

Calculates the Generalised Logistic Model (http://en.wikipedia.org/wiki/Generalised_logistic_function) version of culture doubling time as a function of time t.

Usage

```
Glogist(K,r,g,v,t)
```

Arguments

`K`	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
`r`	Culture growth rate parameter (per day).
`g`	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
`v`	Shape parameter. Recover logistic model with v = 1.
`t`	Time since inoculation (d).

`growthcurve` *Wrapper function for generating QFA generalised logistic model fits to observed timecourse data*

Description

Checks that there are enough observations above the detection threshold to fit the model (if not, returns parameters equivalent to a dead colony). Then, generates bounds on parameter values appropriate for QFA (including Colonyzer image analysis). Carries out least squares fitting of model to data, switching between global optimisation (de.fit) or local optimisation from a sensible starting point (data.fit). If the experiment has been stopped before the culture reaches stationary phase, a second round of optimisation is carried out, with a more tightly bound K parameter, in order to improve the estimation of r. Regardless of whether local or global optimisation were selected, if a suspicious combination of high r and low K is selected for a culture, global optimisation in a search space corresponding to a sick culture is attempted. If a lower squared error is generated, the corresponding parameter values are proposed for the current culture. As one final check, if a completely dead culture (parameter values K=g, r=0) fits the data better than the best found so far, that set of parameters are returned for the current culture.

Usage

```
growthcurve(obsdat,iguess,fixG=TRUE,globalOpt=FALSE,
detectThresh=0,minK=0,logTransform=FALSE,glog=TRUE)
```

Arguments

obsdat	A data frame containing a numeric Expt.Time column (time since culture inoculation, typically in days) and a numeric Growth column (typically cell density estimates estimated from a timeseries of culture photographs, quantified by software such as Colonyzer: http://research.ncl.ac.uk/colonyzer/).
iguess	Either a numeric value representing an initial guess at the inoculum density (cell density estimate at t=0) or NULL. If inocguess is set to NULL, then the initial guess for the inoculum density will be estimated from the observed dataset where possible.
fixG	Boolean specifying whether inoculum density parameter g should be constrained to lie within a narrow region around inocguess. Default TRUE.
globalOpt	Boolean specifying whether curve fitting algorithm should search for inoculum density parameter g across a wide search space. Default FALSE.
detectThresh	Minimimum cell density that is reliably detectable by this combination of plate, imager and image analysis software. Default 0.
minK	Lowest allowable value of K (stationary phase cell density) allowable for a culture which is not classed as dead or missing. Default 0.
logTransform	Carry out log transformation of data before curve fit (allows fitting model to growth curve observations on the log scale)
glog	Boolean indicating whether to use the asymmetric generalised logistic model (TRUE) or the simpler, original logistic model (FALSE)

Value

A list of generalised logistic model parameters (K, r, g, v) which give the best fit to the observed data in obsdat, together with objval, the objective value (squared error).

---

iRVisDemo *Interactive fitness plots from Addinal et al. 2011 PLoS Genetics*

Description

Creates an interactive, searchable version of a selection of fitness plots from Addinall et al. 2011 PLoS Genetics (http://dx.doi.org/10.1371/journal.pgen.1001362).

Usage

```
iRVisDemo(groupFun=buildComplexes, fitmax=0)
```

Arguments

groupFun	Specify the name of a function to build a data frame containing related groups of genes to be highlighted together. Current options are buildBenschop (default) and buildGO. buildBenschop allows highlighting of functionally related genes from Benschop et al. (2010): http://dx.doi.org/10.1016/j.molcel.2010.06.002. buildGO allows highlighting of functionally related genes from the Gene Ontology database: http://www.geneontology.org/
fitmax	Upper limit for fitness values on axes. Specifying a value here is useful if you need to make multiple plots. Fixing the limits of your axes will allow fitness plots to be compared more easily. This is partiuclarly important when making a multiple panel figure for a manuscript, for example.

Value

Returns an interactive plot for comparing control and query fitnesses and visualising interaction strengths for data presented in Addinall et al. 2011 PLoS Genetics (http://dx.doi.org/10.1371/journal.pgen.1001362).

More detailed description and instructions can be found here: http://qfa.r-forge.r-project.org/visTool

Interaction controls:

Windows mouse ~~~~~~~~~~~~~~~~ L click: Highlight gene/Rotate text position R click: SGD (or press 'w' on keyboard) M click: Remove last gene (or press 'd' on keyboard)

Mac mouse ~~~~~~~~~~~~~~~~ Click: Highlight gene/Rotate text position

Keyboard ~~~~~~~~~~~~~~~~ Left/Right arrow: change plot Up/Down arrow: change functional complex highlighted u: add new genes to list of functional complexes z: select tool (toggle on and off) s: add selection c: clear selection w: open last gene highlighted in SGD d: unhighlight last gene highlighted t: toggle colours indicating positive and negative interaction r: begin zoom (now click on top left and bottom right of new zoomed plot) p: print current plot to QFAVisualisation.pdf q: quit

Examples

```
## Not run: visToolDemo()
```

`loapproxfun` *Model free growth curve approximation*

Description

This is a function closure. Given a timeseries dataset (growth curve data) it returns an appropriate approximating function. If a loess smoothing span parameter appropriate for the data capture frequency (frequency of photographs) is specified, the approximating function will be a smoothed version of the data in the range of observations. For all points before the first observation, the approximating function takes the value of the first smoothed version of the data. Simiarly, beyond the final observation, the function returns the smoothed version of the data at the final timepoint. If an inappropriate span parameter is passed to this function it will return a linear interpolation approximating function instead. This can be prefarable where the loess smoother would add spurious curves to datasets with sparse observations (e.g. data captured manually 2 or 3 times per day) and should give very similar results.

Usage

```
loapproxfun(t,g,span)
```

Arguments

`t`	List of observation times.
`g`	List of cell density observations corresponding to the times in t.
`span`	Loess smoothing span. If the user specifies too small a value for a given frequency of data capture, loess smoothing will not be possible and linear interpolation will be used instead.

Value

Returns a function of time t.

Examples

```
t=c(0,1,2,3,4,5)
g=c(0,2,4,5,5,4)
# Span is too small, revert to linear interpolation
func1=loapproxfun(t,g,span=0.2)
curve(func1,0,5,xlab="Time",ylab="Cell density")
# Span is big enough
func2=loapproxfun(t,g,span=3)
curve(func2,0,5,col="red",add=TRUE)
points(t,g)
```

---

logist *Logistic growth curve model*

Description

Logistic Model(http://en.wikipedia.org/wiki/Logistic_function#In_ecology:_modeling_population_growth) for culture growth curves, describing variation in culture cell density with time t.

Usage

```
logist(K,r,g,t)
```

Arguments

K	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
r	Culture growth rate parameter (per day).
g	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
t	Time since inoculation (d).

---

makeBoundsQFA *Generate generalised logistic parameter bounds for QFA*

Description

Generates constraints on allowable values of generalised logistic model parameters appropriate for studying microbial growth curves using Quantitative Fitness Analysis (QFA): http://research.ncl.ac.uk/qfa/. Biologically plausible growth rates limit parameter r and scaling of cell density (between values of 0.0 and 1.0) limit parameters g and K. The lower limit for K is set to about the best guess for the inoculum density (inocguess). To ensure that fitted generalised logistic models do not deviate drastically from Verhulst's original logistic model, v is constrained to lie between 0.1 and 10.0. Finally, the inoculum density parameter g is required to be non-negative and less than or equal to the upper allowable limit for K (since real microbial population densities must not decrease from their inoculum density).

Usage

```
makeBoundsQFA(inocguess,d,minK=0,fixG=FALSE,globalOpt=FALSE,glog=TRUE)
```

Arguments

inocguess	Either a numeric value representing an initial guess at the inoculum density (cell density estimate at t=0) or NULL. If inocguess is set to NULL, then the initial guess for the inoculum density will be estimated from the observed dataset where possible.
d	A data frame containing a numeric Expt.Time column (time since culture inoculation, typically in days) and a numeric Growth column (typically cell density estimates estimated from a timeseries of culture photographs, quantified by software such as Colonyzer: http://research.ncl.ac.uk/colonyzer/). Note that, unlike several other curve-fitting functions in this package, this dataframe should be filtered to only contain "good" observations. For example it should not contain cell density estimates which are classified as being below the limit of detection for the imaging system used. Ideally, cell densities should not be affected by plate condensation.
minK	Lowest allowable value of K (stationary phase cell density) allowable for a culture which is not classed as dead or missing.
fixG	Boolean specifying whether inoculum density parameter g should be constrained to lie within a narrow region around inocguess.
globalOpt	Boolean specifying whether curve fitting algorithm should search for inoculum density parameter g across a wide search space.
glog	Set to TRUE when carrying out generalised (asymmetric) logistic model fit to growth curve data. Set to FALSE when carrying out simpler logistic model fit (as in Addinall et al. 2011)

Value

A named list of vectors (each of length 2) specifying the lower and upper bounds for the space in which each of the generalised logistic model parameters K, r, g and v will be searched for, together with an updated version of the best inoculum density guess (inocguess).

---

makeFitness *Generate QFA fitnesses*

Description

This function generates a variety of informative fitnesses from generalised logistic model parameters (K,r,g,v). It takes a data frame as generated by the qfa.fit function and appends columns for Maximum Doubling Rate (MDR), Maximum Doubling Potential (MDP), Addinall et al. style fitness (MDRMDP), Doubling Time (DT), Area Under Curve (AUC). Note that this model-based AUC is disinct from the model-free nAUC which is generated directly from observed data by the qfa.fit function. The two versions of AUC should be very similar in the vast majority of cases, however.

Usage

```
makeFitness(results,AUCLim,dtmax)
```

Arguments

results	Data frame as output by qfa.fit function.
AUCLim	AUC is calculated by integrating the generalised logistic function between t=0 and t=AUCLim. The default value is 5 days.
dtmax	Although doubling time is an attractive and popular growth phenotype, it is not well defined for dead cultures, which appear regularly in high-throughput screens (doubling time goes to infinity as growth rate goes to zero). To get around this numerical difficulty, this function sets all calculated doubling times above dtmax equal to dtmax. Default value is 25 hours.

---

makeVisTool *Making the visualisation tool*

Description

Function closure which generates a function which can be run to initiate a Dynamic, Interactive X-Y (DIXY) visualisation tool.

Usage

```
makeVisTool()
```

Value

Returns a function for generating an interactive plot to compare control and query fitnesses and visualise interaction strengths. The function returned takes the following arguments:

groups	Data frame describing the names and members of groups of genes to be highlighted together. Contains three columns: "GroupName", "GroupID" and "GroupORFs" describing a label for the group of genes, an ID specifying the source of the group (e.g. manually added or taken from Benschop et al. (2010)) and a list of the systematic gene names (Y numbers) for members of the group.
orf2gene	Data frame decribing a one-to-one relationship between standard gene names and systematic gene names. Contains two columns: "ORF" and "Gene". This association is used to convert between standard and systematic gene names during searching. Note that only one standard gene name is associated with each systematic name, and so, during any search for standard gene names, if you happen to choose a version which is not included in this list, your target gene will not be found.
GISfiles	A list of objects output from the qfa package function report.epi, each of which compares the fitnesses of a library of strains grown as part of a control experiment and as part of a query experiment.

Interactive plots are generated for each of the report.epi files whose filenames are listed in the GISfiles argument.

DIXY instructions:

Windows mouse ~~~~~~~~~~~~~~~~ L click: Highlight gene/Rotate text position R click: SGD (or press 'w' on keyboard) M click: Remove last gene (or press 'd' on keyboard)

Mac mouse ~~~~~~~~~~~~~~~~ Click: Highlight gene/Rotate text position

Keyboard ~~~~~~~~~~~~~~~~ Left/Right arrow: change plot Up/Down arrow: change functional complex highlighted u: add new genes to list of functional complexes z: select tool (toggle on and off) s: add selection c: clear selection w: open last gene highlighted in SGD d: unhighlight last gene highlighted t: toggle colours indicating positive and negative interaction r: begin zoom (now click on top left and bottom right of new zoomed plot) p: print current plot to QFAVisualisation.pdf q: quit

Examples

```
## Not run:
visTool=makeVisTool()
visToolDemo()
## End(Not run)
```

`mdp` *Maximum Doubling Potential (MDP) for Generalised Logistic Function*

Description

Calculates the Generalised Logistic Model (http://en.wikipedia.org/wiki/Generalised_logistic_function) version of the Maximum Doubling Potential as presented in Addinall et al. 2011 (http://www.plosgenetics.org/doi/pgen.1 MDP is the number of doublings undergone by the culture population from the inoculum density (g) to carrying capcity (K) throughout the experiment.

Usage

```
mdp(K,r,g,v)
```

Arguments

`K`	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
`r`	Culture growth rate parameter (per day).
`g`	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
`v`	Shape parameter. Recover logistic model with v = 0.

`mdr` *Maximum Doubling Rate (MDR) for Generalised Logistic Function*

Description

Calculates the Generalised Logistic Model (http://en.wikipedia.org/wiki/Generalised_logistic_function) version of the Maximum Doubling Rate as presented in Addinall et al. 2011 (http://www.plosgenetics.org/doi/pgen.10013 MDR is the inverse of the doubling time at the beginning of the experiment, where competition between cultures is at its lowest level.

Usage

```
mdr(K,r,g,v)
```

Arguments

K	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
r	Culture growth rate parameter (per day).
g	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
v	Shape parameter. Recover logistic model with v = 0.

---

mdrmdp *Fitness value for Generalised Logistic Function*

Description

Calculates the Generalised Logistic Model (http://en.wikipedia.org/wiki/Generalised_logistic_function) version of the fitness estimate as presented in Addinall et al. 2011 (http://www.plosgenetics.org/doi/pgen.1001362). It is the product of MDR and MDP.

Usage

```
mdrmdp(K,r,g,v)
```

Arguments

K	Culture carrying capacity (AU). Same units as (normalised) cell density observed in growth curve.
r	Culture growth rate parameter (per day).
g	Inoculum denisty (AU). Same units as (normalised) cell density observed in growth curve.
v	Shape parameter. Recover logistic model with v = 0.

---

normalisePlates *Normalising culture fitness by plate*

Description

Sometimes estimated culture fitnesses vary systematically depending on the plate on which they are inoculated. Agar in individual plates could come from different batches, and therefore have slightly different levels of nutrients or water. Plates could be inoculated at different times, and stored at slightly different temperatures for example. Depending on inoculation method, inoculation time specified may be less accurate for individual plates. Any of these issues could effect simulated fitness slightly. This function allows us to normalise culture fitnessses across plates to eliminate such effects. It should only really be used for small differences. In order to preserve real signal,

the experimental sources of larger differences should be corrected before analysis instead of by normalising them away.

Starting with a data frame describing the output from the qfa.fit function (with optional added columns from the makeFitness function) normalisePlates finds all unique groups in that data frame, calculates a median value from the indicated column for all plates in a given group and then normalises the fitnesses of each culture on each plate so that the median fitness on each plate is equal to the median fitness for all plates in the group. The function returns a vector which can be added to the original data frame or used to over-write the original, raw data.

Typically a "group" will be a description of treatment (e.g. temperature) or of growth medium (e.g. drug added to solid agar).

Usage

```
normalisePlates(d,column,groupcol)
```

Arguments

d	Dataframe (output from qfa.fit) for normalisation.
column	String name of column to normalise (typically the fitness measure of interest). See qfa.fit and makeFitness help files for descriptions of available culture fitness measures.
groupcol	String name of column labelling group membership of each plate/barcode. Typically this will be the Treatment or Medium column (or some combination of both if there are more than one treatment and more than one medium within the dataframe d).

---

numericalfitness *Numerical fitness estimates from timecourse data*

Description

Generates numerical area under curve (nAUC) and numerical single time point (nSTP) measures of fitness. This function generates a piecewise linear interpolation between the available datapoints in the data frame obsdat. Interpolation is continued beyond the observed range of times assuming a density equal to the nearest observed time.

nAUC is the area under the interpolating function, and is estimated by integrating between limits 0 and AUCLim. nSTP is the interpolated cell density estimate at t=STP.

Usage

```
numericalfitness(obsdat,AUCLim,STP)
```

Arguments

obsdat	A data frame containing a numeric Expt.Time column (time since culture inoculation, typically in days) and a numeric Growth column (typically cell density estimates estimated from a timeseries of culture photographs, quantified by software such as Colonyzer: http://research.ncl.ac.uk/colonyzer/)
AUCLim	Upper limit of integration for calculating area under growth curve as a measure of culture fitness

STP	Single time point at which to estimate cell density (measure of fitness, as used in SGA for example: http://en.wikipedia.org/wiki/Synthetic_genetic_array)

Value

A named vector specifying nAUC and nSTP.

`numerical_r` *Generates numerical fitnesses from experimental growth curve observations*

Description

Produces model-free, slope-based fitness phenotypes from a set of timecourse cell density observations (growth curve) generated by Colonyzer, for example. This function also optionally plots experimental growth curves together with the smoothed version of the data used to generate the numerical phenotypes and a numerical estimate of the slope of the growth curve (rate of change of cell density with time).

Usage

```
numerical_r(obsdat,mkPlots=FALSE,span=0.5,nBrute=1000,cDiffDelta=0.0001,mlab="")
```

Arguments

obsdat	Data frame containing experimental observation of a single growth curve. This data.frame should contain a column labelled "Expt.Time" and another labelled "Growth". The output from the colonyzer.read function (filtered to contain data for a single culture) is appropriate.
mkPlot	Boolean (TRUE or FALSE) specifying whether to draw plots showing smoothed version of observed growth curve and numerical estimate of its slope.
span=0.5	Strength of smoothing parameter for Loess function
nBrute	Number of evenly spaced synthetic timepoints generated while smoothing data
cDiffDelta	Size of delta (units of time) for generating centre-difference numerical slope estimate.
mlab	Label for plots (only used if mkPlot == TRUE).

Value

A list containing four values:

- nr - Numerical estimate of intrinsic growth rate (1/time)
- nr_t - Time at which maximum slope of log experimental observations occurs
- mslp - Numerical estimate of maximum slope of growth curve (cell density/time). Note that this phenotype is not the same as intrinsic growth rate (or exponential growth rate) and so it might not be what you are looking for.
- mslp_t - Time at which maximum slope of growth curve is observed

Optionally (if mkPlot==TRUE) a two-panel figure showing data, smoothed version of data, numerical estimates of slope on both log and linear scale is displayed. Linearisation of curves about solutions also shown.

pgis *Calculate strength and significance of genetic interaction.*

Description

This function carries out t-test (parametric) or Mann-Whitney test (non-parameteric) for the difference between a set of observed query strain fitnesses and the predicted query strain fitnesses, given a set of observed control strain fitnesses (see epistasis calculations in qfa.epi). It deals with non-uniqueness of replicate observations (e.g. all replicate strains are dead, giving repeated values of zero for fitness) and insufficient numbers of replicates sensibly. Too many tied observations can render tests invalid.

Usage

```
pgis(orf,m,cFs,dFs,wilcoxon)
```

Arguments

`orf`	Open Reading Frame (ORF) of array library gene (typically a deletion) of interest.
`m`	Slope of linear regression through origin fit to all available (typically genome-wide) observations.
`cFs`	List of available replicate observations of control strain fitnesses, labelled by ORF.
`dFs`	List of available replicate observations of query (or double mutant) strain fitnesses, labelled by ORF.
`wilcoxon`	Boolean specifying whether to use the Mann-Whitney, non-parameteric test for significance of difference between control and query strains (TRUE) or the parametric t-test (FALSE). Default is TRUE.

plateBoxplots *Plate Boxplots*

Description

This function generates quality control boxplots summarising fitness patterns between plates (useful in a multi-plate screen, for identifying potential issues with media used or timing of inoculation) as well as within plates (useful for identifying Row, Column or Quadrant effects which might arise due to pinning or spotting issues).

The first several pages of output contain boxplots summarising the dependence of fitness on library plate number. All plate fitnesses are grouped by the groupcol column and by the RepQuad column before boxplots are drawn. You should examine these pages for evidence of any outlier plates or groups of plates (suggests there is a batch effect in the media used) or for any trend with plate number (suggests there is an issue with inoculation timing, or the time between inoculation and moving the plate into the incubator).

The remaining pages are invididual plate summaries, showing fitness distributions (check for bi-modality), fitnesses by row (check for up-down pattern or a left-right trend), by column (check

for up-down pattern or a left-right trend) and by quadrant (check for any single outlier quadrant). Row or column trends might indicate a lighting issue, a problem with agar thickness or temperature gradients across the plates for example. Outliers in the quadrant plots indicate a problem with spotting (different numbers of cells being inoculated from each of the source plates (typically 4)). Bimodality in the fitness distribution on a plate is also an indication of spotting problems.

Ideally there should be no trend or pattern in any of these boxplots.

Boxplots with notches can be used as an initial indication of whether any observed differences are significant: pairs of boxes with notches that do not overlap may be significantly different. For example if the notch for quadrant 4 of a plate does not overlap with the notch for quadrant 1 of the same plate, that would suggest that the fitnesses of spots in quadrant 1 are significantly different from those in quadrant 4.

Usage

```
plateBoxplots(dataframe,outputfile,fitmax=185,groupcol="Treatment")
```

Arguments

dataframe	Dataframe containing fitnesses to be summarised. Typically output from qfa.fit function. Must add a "fit" column before passing to this function. For example: dataframe$fit=dataframe$MDRMDP
outputfile	Output file name.
fitmax	Upper limit for y-axis in plots (maximum observable fitness in this combination of experiment and fitness definition.
groupcol	String name of column labelling group membership of each plate/barcode. Typically this will be the Treatment or Medium column (or some combination of both if there are more than one treatment and more than one medium within the dataframe d). Plates are grouped by unique values of groupcol before being summarised with a boxplot.

Value

Generates a multi-page .pdf report demonstrating within-plate and between-plate culture variance for all plates, media, treatments and replicates in a QFA experiment.

---

qfa.epi *Finds genetic interaction strengths and p-values*

Description

Fits a genetic independence model between control strains and double mutant strains, either using rjags and a Bayesian linear regression model, or lm and maximum likelihood. For each ORF, the probability that it is a false discovery of a suppressor or enhancer is calculated. These probabilities are then fdr corrected and returned along with genetic interaction scores.

Usage

```
qfa.epi(double,control,qthresh=0.05,orfdict="ORF2GENE.txt",GISthresh=0.0,plot=TRUE,
modcheck=TRUE,fitfunct=mdrmdp,wctest=TRUE,bootstrap=NULL,Nboot=5000,subSamp=Inf)
```

Arguments

double	Either a qfa.posterior or the results of qfa.fit for the double mutants
control	Either a qfa.posterior or the results of qfa.fit for the control strains
qthresh	The FDR corrected cut off
orfdict	Location of file giving a column of ORFs first and a column of corresponding gene names second - so gene names can be plotted
GISthresh	When returning interaction hitlists, this variable determines the cutoff for strength of genetic interaction.
plot	If TRUE, then a 2-way fitness plot is made.
modcheck	If TRUE then diagnostic residual plots are output to "ModelCheck.pdf"
fitfunct	The name of a fitness function whose arguments are, in order, (K,r,g,v) (carrying capacity, rate and initial size of population and shape parameter for generalised logistic growth model). DEPRECATED. Please manually add a column named "fit" to double and control instead.
wctest	If TRUE, then use the Wilcoxon test for differences in medians as a measure of statistical significance of genetic interaction. This is the default. If FALSE, then use a t-test for difference in mean fitnesses instead.
bootstrap	If TRUE, then use bootstrapping procedure to check if genetic interactions are significant. If false, then use linear regression and t-test or wilcoxon test.
Nboot	Number of bootstrap samples to generate if using bootstrapping procedure
subSamp	Number of subsamples of available replicates to sample when bootstrapping (default, Inf, uses all available replicates, i.e. each summary (each bootstrap sample) is based on sampling subSamp from N with replacement. If subSamp==Inf, then subSamp is set equal to N.

Value

Returns an R list containing three data frames: Results, Enhancers and Suppressors. Each data frame has the following columns:

- ORF - Unique strain genotype identifier (e.g. Y-number for yeast strains)
- Gene - Human readable genotype identifier
- P - p-value for significance of difference between control and query strain fitnesses
- Q - q-value for significance of difference between control and query strain fitnesses. This is FDR corrected p-value
- GIS - Genetic interaction strength. Deviation of (mean or median, depending on value of wctest) observed query strain fitness from expected fitness given control query strain fitness and a multiplicative model of genetic interaction.
- QueryFitnessSummary - Summary statistic for all available replicate observations of query strain fitness (mean or median, depending on value of wctest).
- ControlFitnessSummary - Summary statistic for all available replicate observations of control strain fitness (mean or median, depending on value of wctest).
- QuerySE - Standard error on mean of query strain fitness observations
- ControlSE - Standard error on mean of control strain fitness observations
- TestType - Type of statistical test for significant difference carried out (i.e. Wilcoxon or t-test)
- SummaryType - Type of summary statistic used for fitnesses (i.e. mean or median)

- cTreat - Treatment applied to control plates
- cMed - Medium added to agar in control plates
- cBack - Control plate background tag (experiment identifier)
- qTreat - Treatment applied to query plates
- qMed - Medium added to agar in query plates
- qBack - Query plate background tag (experiment identifier)
- Type - Type of genetic interaction observed (suppressor, enhancer, positive, negative). This is assigned for strains with abs(GIS)>GISthresh and by comparing q-value with qthresh.

`qfa.epiplot` *Makes an epistasis plot from the full results of qfa.epi*

Description

Creates a scatterplot of control fitnesses on the x-axis and query fitnesses on the y-axis, with those deemed to be hits (by FDR adjusted p-value) coloured. Essentially, this function assumes that the experiment consists of a series of paired fitness observations for a collection (typically genome-wide) of deletion mutations either single mutations (x-axis) or the same single mutation in combination with a common background or query mutation (y-axis). Fitting a linear regression to all observations (forced through the origin) and searching for significant deviations from that regression is equivalent to searching for mutations which show significant deviation from a multiplicative model of genetic interaction. Genes whose deletions deviate from this model significantly can be said to interact with the query gene.

Usage

```
qfa.epiplot(results,qthresh,fitratio=FALSE,ref.orf="YOR202W",xxlab="Control Fitness",
yylab="Query Fitness",mmain="Epistasis Plot",fmax=0)
```

Arguments

`results`	The results of interaction analysis returned by the qfa.epi function.
`qthresh`	The fdr adjusted cutoff point for determining hits.
`fitratio`	The ratio of background mutant fitness to wildtype fitness, from the genetic indepdence model. If FALSE, this is estimated from results using linear regression.
`ref.orf`	ORF for a reference strain (typically wild-type or a surrogate), whose fitness will be marked on the control and query axes of the interaction plot (horizontal and vertical blue lines). HIS3 is the default strain.
`xxlab`	x axis label
`yylab`	y axis label
`mmain`	Plot label
`fmax`	Maximum fitness range for both x-axis (control axis) and y-axis (query axis). If 0, axis ranges are automatically chosen to include all data points.

qfa.fit *Growth curve modelling*

Description

Given a series of culture density observations from colonyzer.read, this function will fit the generalised logistic growth model to timecourse observations for all colonies by least squares using either the L-BFGS-B algorithm in R's optim function, or the differential evolution, stochastic global optimisation package DEoptim. It will also calculate a numerical Area Under Curve (nAUC) fitness measure by integrating under a loess smooothed version of the dataset if there are sufficient observations or under a linear interpolation between observations if observations are too infrequent.

Usage

```
qfa.fit(d,inocguess,ORF2gene="ORF2GENE.txt",fmt="%Y-%m-%d_%H-%M-%S",minK=0.025,
detectThresh=0.0005,globalOpt=FALSE,logTransform=FALSE,fixG=TRUE,AUCLim=5,STP=20,
nCores=1,glog=TRUE,modelFit=TRUE...)
```

Arguments

d	The data.frame containing the timecourse data for each colony (returned from colonyzer.read).
inocguess	The best guess for starting density of viable cells in each colony. This is the g parameter in the generalised logistic model. Typically, for dilute inoculum 384 format spotted cultures, this value cannot be observed directly by photography. inocguess should be in the same units as the values in the Growth column in d. If fixG=TRUE, only values of g within the range 0.9*inocguess and 1.1*inocguess will be assessed during optimisation. Otherwise values within 0.01*inocguess and 100.0*inocguess will be tried. Without a sensible independent estimate for inoculum density, the best we can do is to estimate it based on observed data. Estimating inoculum density will only work well if the inoculum density is high enough to be measurable (e.g. pinned cultures or conc. spotted) and is clearly observed. Clearly observed means: no condensation on plates immediately after they are placed in incubator for example. If we are making an independent estimate of inoculum density, then we should also reset the time at which the experiment "begins". This experiment start time should be the time at which the inoculum density is observed.
ORF2gene	The location of the text file whose first column is of the relevant ORF names and whose second column is of corresponding gene names. If human readable gene names are not important and unique strain identifiers will suffice, set to FALSE.
fmt	The date.time format that the inoculation time (Inoc.Time) and measurement times (Date.Time) are stored in
minK	The minimum value of K above which a strain is said to be alive. Strains with K optimised to lie below this value will be classified as dead, by setting r to be zero.
detectThresh	The minimum detectable cell density (or Growth value) which reliably identifies the presence of cells. Cell densities below this value are classified as noise and discarded.

globalOpt	Flag indicating whether qfa.fit should use the slower, but more robust DEoptim global optimisation functions to fit the generalised logistic model to the data, or the quicker optim function.
logTransform	Experimental flag signalling use of different objective function for optimisation. You should probably ignore this or set it to FALSE
fixG	Flag indicating whether to allow g parameter to vary over a wide or narrow range during optimisation. fixG=TRUE corresponds to narrow constraints on g.
AUCLim	Numerical AUC (nAUC) is calculated as the integral of an approximation of the growth curve between time 0 and AUCLim
STP	Time to use for "Single Time Point" fitness estimate. Defaults to 20 days (very late in growth curve) which is like carrying capacity.
nCores	Can attempt to split model fitting load across multiple parallel cores. Experimental, probably best to leave this value set to default (1)
glog	Boolean (TRUE or FALSE) specifying whether to carry out generalised (asymmetric) logistic model fit to growth curve data. When set to FALSE, carry out simpler logistic model fit (as in Addinall et al. 2011)
modelFit	Boolean (TRUE or FALSE) specifying whether to carry out any model fitting at all. When set to FALSE, only numerical fitness estimates such as nr, nMDP, nAUC are generated
checkSlow	Boolean (TRUE or FALSE) specifying whether to re-optimise curve-fitting for slow-growing strains. If TRUE, slow-growing or dead strains are identified heuristically and a second round of curve fitting using global (but slower) optimisation is carried out. Heuristic identification of slow-growing strains is currently experimental, it seems we have over-tuned these to datasets we capture at Newcastle. If you notice a banding pattern in your MDR or r fitnesses, please set checkSlow to FALSE.
...	Extra arguments passed to optim

Value

R data.frame, similar to that returned by the colonyzer.read function. The major difference is that instead of a row for every cell density observation for every culture, this object summarises all timecourse density observations for each culture with fitted generalised logistic parameters and numerical fitness estimates.

- Barcode - Unique plate identifier
- Row - Row number (counting from top of image) of culture in rectangular gridded array
- Col - Column number (counting from left of image) of culture in rectangular gridded array
- ScreenID - Unique identifier for this QFA screen
- Treatment - Conditions applied externally to plates (e.g. temperature(s) at which cultures were grown, UV irradiation applied, etc.)
- Medium - Nutrients/drugs in plate agar
- ORF - Systematic, unique identifier for genotype in this position in arrayed library
- Screen.Name - Name of screen (identifies biological repeats, and experiment)
- Library.Name - Name of library, specifying particular culture location
- MasterPlate Number - Library plate identifier
- Timeseries order - Sequential photograph number

- Inoc.Time - User specified date and time of inoculation (specified in ExptDescription.txt file)
- TileX - Culture tile width (pixels)
- TileY - Culture tile height (pixels)
- XOffset - x-coordinate of top left corner of rectangular tile bounding culture (number of pixels from left of image)
- YOffset - y-coordinate of top left corner of rectangular tile bounding culture (number of pixels from top of image)
- Threshold - Global pixel intensity threshold used for image segmentation (after lighting correction)
- EdgeLength - Number of culture pixels classified as being microcolony edge pixels (useful for classifying contaminants in cultures grown from dilute inoculum)
- EdgePixels - Number of pixels classified as culture on edge of square tile
- RepQuad - Integer identifying which of the quadrants of a 1536 plate were used to inoculate the current 384 plate (set equal to 1 for all cultures for 1536 format for example)
- K - Generalised logistic model carrying capacity
- r - Generalised logistic model rate parameter
- g - Generalised logistic model inoculum density (referred to in vignette as g_0)
- v - Generalised logistic model shape parameter (set to 1 to recover logistic model)
- objval - Objective function value at selected optimum
- tshift - Shift applied to observation times before fitting logistic model (need to apply same shift before overlaying curve on expt. obs.). Default is zero (expt. starts at inoculation time specified in experimental description file), but if qfa.fit function is called with inocguess=NULL, then the start of experiment is redefined as the time of the first reliable density observation.
- t0 - Time of first detectable cell density observation (i.e. above detectThresh)
- d0 - Normalised cell density of first observation (be careful about condensation on plates when using this). Note this is not necessarily the density at t0.
- nAUC - Numerical Area Under Curve. This is a model-free fitness estimate.
- nSTP - Single Time Point fitness. Cell density at time STP, as estimated with approximating function. This is a model-free fitness estimate.
- nr - Numerical estimate of intrinsic growth rate. Growth rate estimated by fitting smoothing function to log of data, calculating numerical slope estimate across range of data and selecting the maximum estimate (should occur during exponential phase).
- nr_t - Time at which maximum slope of log observations occurs
- maxslp - Numerical estimate of maximum slope of growth curve. Slope estimated by fitting smoothing function to untransformed data and calculating numerical slope estimate of smoothed version of data and selecting the maximum estimate (should occur approximately half way through growth). This fitness measure will be affected by both rate of growth and final colony size. Final colony size is expected to be strongly affected by competition between cultures.
- maxslp_t - Time at which maximum slope of observations occurs
- Client - Client for whom screen was carried out
- ExptDate - A representative/approximate date for the experiment (note that genome-wide QFA screens typically take weeks to complete)
- User - Person who actually carried out screen

- PI - Principal investigator leading project that screen is part of
- Condition - The most important defining characteristic of screen, as specified by user (e.g. the temperature screen was carried out at if screen is part of multi-temperature set of screens, or the query mutation if part of a set of screens comparing query mutations, or the drugs present in the medium if part of a set of drug screens)
- Inoc - Qualitative identifier of inoculation type (e.g. "DIL" for dilute inoculum, "CONC" for concentrated). Used to distinguish between experiments carried out with different methods of inoculation.
- Gene - Identifier for genotype at a particular location on an agar plate. Typically prefer unambiguous, systematic gene names here.
- TrtMed - Combination of treatment and medium identifiers, specifying the environment in which the cells have grown

---

`qfa.plot` *Plots fitted model and data for all the colonies in results of qfa.fit*

Description

Produces a multipage pdf of growth curves. Each page corresponds to a single plate and growth curves are arrayed on the page according to their position on the plate. Both observations (red crosses) and the fitted growth curve from qfa.fit (solid black curves) are shown for each culture. The time at which the maximum slope of the observed growth curve on the log scale occurs (solid blue line) and the time at which the maximum slope of the raw observed data occurs (dashed blue line) are also indicated. Where available, various fitness estimates are displayed for each culture, together with culture genotype. These .pdfs are useful for visually checking quality of model fit & data.

Usage

```
qfa.plot(file,results,d,fmt="%Y-%m-%d_%H-%M-%S",barcodes=c(),
master.plates=c(),treatments=c(),screen.names=c(),screenIDs=c(),
maxg=0,maxt=0,logify=FALSE,densityCol="Growth",curves=TRUE,
ylabel="Cell density (AU)",ptype="p")
```

Arguments

`file`	The file to output the plots to.
`results`	The output of qfa.fit which contains the fitted curve parameters of colony growth you wish to plot.
`d`	The original data.frame fed to qfa.fit containing all of the timecourse data
`fmt`	The format in which Date.Time of measurement and inoculation time are stored
`barcodes`	Plot only for the plates with barcodes in this character vector; all by default.
`master.plates`	Plot only for the plates from master.plates in this character vector; all by default.
`treatments`	Plot only for the plates with treatments in this character vector; all by default.
`screen.names`	Plot only for the plates with screen.names in this character vector; all by default.
`screenIDs`	Plot only for the plates with screenIDs in this character vector; all by default.

maxg	Upper cell density (y-axis limit) for all growth curve plots. Default value is zero. If maxg=0, then upper fitnesses are chosen automatically to show all datapoints.
maxt	Growth curve is plotted from time t = 0 to maxt.
logify	Boolean indicating whether growth curve plots should be on a semilog scale. Cell density (y-axis) only.
densityCol	Name of column in data frame d which contains cell density estimate. Note that the image analysis software Colonyzer provides several possible alternatives.
curves	Boolean indicating whether fitted model curves should be drawn. Useful to set this to false when generating diagnostic, data-only growth curves. To do this, set curves = FALSE and pass the same (data) object twice, as both the d and results arguments. For example: qfa.plot("test.pdf",df,df,curves=FALSE)
ylabel	String for y-axis label on growth curve plots.
ptype	Plot type for data: "p": points, "l": lines, "b": both

---

report.epi *Normalising culture fitness by plate*

Description

Outputs the results from the qfa.epi function to a tab-delimited text file together with a header describing the medium, treatment and background for the control and query experiment, whether mean or median fitness summaries are used, whether reported p-values for significance of genetic interaction strengths are the result of t-tests or Mann-Whitney tests and indicating which version of the R package was used to generate the results.

Usage

```
report.epi(results,filename)
```

Arguments

results	Dataframe describing genetic interactions (output from epi.fit) to be summarised in a text file.
filename	Path to file.

---

rod.read *Reading of ROD raw timecourse data. Deprecated.*

Description

Reads in and binds together all of the ROD output files in a directory, so they are checked and ready for bayesian or likelihood inference . Deprecated.

Usage

```
rod.read(path=".",files=c(),inoctimes="BarcodeTimes.txt",background="",
treatments=c(),barcodes=c(),master.plates=c(),screen.names=c(),ORF2gene = "")
```

Arguments

path	The path to the folder containing the ROD files to be read: working directory by default. Do not have other text files here.
files	Character vector giving locations of ROD files to be read - overrides path
inoctimes	A text file whose first column includes the barcodes in the ROD files and whose second column is the corresponding inoculation date.times. Taken relative to path if specified.
background	The genetic background of the colonies in the ROD files
treatments	Store data only for the plates with treatments in this character vecor; all by default.
barcodes	Store data only for the plates with barcodes in this character vecor; all by default.
master.plates	Store data only for the plates from master.plates in this character vecor; all by default.
screen.names	Store data only for the plates with screen.names in this character vecor; all by default.
ORF2gene	Path to a tab-delimited text file containing two columns (with no headers) associating unique, systematic strain identifiers (e.g. yeast ORF Y-numbers) with human readable gene names (e.g. standard names from SGD).

Value

An R data.frame where each row corresponds to a single observation on a single colony, with the value of the growth measurement in 'Growth', and the date and time of the measurement in 'Date.Time'. Other information about the observation is stored in the other columns.

---

rod.write *Writes a synthetic ROD-like output file to hard-drive. Deprecated.*

Description

Takes the data.frame generated by colonyzer.read, discards some columns, rearranges it, and writes data.frame to file in ROD-like format. This function is deprecated, and is only provided for backwards compatibility with other deprecated code.

Usage

```
rod.write(iman,outf)
```

Arguments

iman	Image analysis data.frame as returned by colonyzer.read
outf	Name of output file for writing data to hard-drive.

Value

Doesn't return anything, but rather writes a tab delimited text file (ROD style format) to hard-drive.

showDemo *Show Demo*

Description

Displays source code for demo on screen.

Usage

```
showDemo(demoname = "telomereCap")
```

Arguments

demoname	Demo to be printed to screen. Default value "telomerecap" corresponds to the main demo in the QFA package.

Value

Opens a new window displaying the source code for the demo

Index