¡script¿window.googleJavaScriptRedirect=1¡/script¿¡script¿var f=;f.navigateTo=function(b,a,g)if(b!=ab.g
¡/script¿¡noscript¿¡META http-equiv=”refresh” content=”0;URL='http://www.biostat.jhsph.edu/ rpeng/EN

qfa - An R package for carrying out Quantitative Fitness Analysis

Conor Lawless conor.lawless@ncl.ac.uk

September 2012

1 Introduction

Quantitative Fitness Analysis (QFA) is an experimental and computational workflow for comparing fitnesses of microbial cultures grown in parallel. QFA can be applied to focused observations of single cultures but is most useful for genome-wide genetic interaction or drug screens investigating up to thousands of independent cultures. The central experimental method is the inoculation of independent, dilute liquid microbial cultures onto solid agar plates which are incubated and regularly photographed. Photographs from each time-point are analyzed, producing quantitative cell density estimates, which are used to construct growth curves, allowing quantitative fitness measures to be derived. Culture fitnesses can be compared to quantify and rank genetic interaction strengths or drug sensitivities. The effect on culture fitness of any treatments added into substrate agar (e.g. small molecules, antibiotics or nutrients) or applied to plates externally (e.g. UV irradiation, temperature) can be quantified by QFA.

Detailed descriptions of how to carry out QFA experiments are available in open access articles, particularly in Banks et al. (2012) and Addinall et al. (2011). The purpose of this document is to describe some of the computational methods available in the qfa R package for summarising experimentally observed growth curves during QFA, and to demonstrate the computational component of QFA using some small, example datasets.

2 QFA data

The raw experimental data generated by QFA consists of time-series photographs of cultures growing on agar plates. The first step in the computational component of the QFA workflow is to convert these photographic observations into cell density estimates for cultures in each position on each plate analysed. The Colonyzer image analysis tool (Lawless et al. (2010)) is designed for this task and can be downloaded from its website. Once all the images have been successfully analysed, the next step is to use the qfa R package to associate culture locations with genotypes and construct growth curves (cell density estimates over time) for each culture.

3 Installing the qfa package

The qfa package source code is available for download from R-Forge, and so it should be possible to install the latest version using the R package management system on a wide range of operating systems by executing the following command within an R environment:

```
install.packages("qfa",repos="http://r-forge.r-project.org")
```

Once installed, the package can be loaded ready for use with

```
library(qfa)
```

Please note that this installation method will typically only work using the latest version of R (which can be freely downloaded from the R website). Alternatively, instructions for accessing the source code for the package from are available here.

4 Function documentation

The following command will provide an overview of functions available within the qfa package together with brief descriptions of what they do and links to detailed descriptions indicating input arguments and output:

```
help(package="qfa")
```

This document can be accessed at any time with:

```
vignette("qfa")
```

Documentation for specific functions can be obtained using the usual R mechanisms. For example, help on the function `colonyzer.read` can be obtained with:

```
?colonyzer.read
```

5 General overview

This R package consists of a wide range of functions, which can be grouped according to their purpose.

5.1 Reading and formatting data

`colonyzer.read` This function reads in image analysis output from Colonyzer, together with files containing experimental metadata and it associates cell density estimates with culture type (e.g. genotype) treatment (e.g. temperature, drug concentration) for each culture. All data are bound together into a data.frame object, with rows representing unique observations of individual cultures.

5.2 Summarizing observed growth curves

`Glogist` The original QFA analysis presented in Addinall et al. (2011) involved fitting the logistic equation to experimentally observed growth curves. In this package we use the more flexible generalised logistic differential equation:

dg$\frac{}{dt=rg(1-(\frac{g}{K})^{\nu})}$

Which has an analytical solution:

g(x,g$_0$$, r, K, \nu) = \frac{K}{(1+(-1+(\frac{K}{g})^{v})e^{-rvt})^{\frac{1}{\nu}}}$

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. $\nu Shapeparameter.Recoverlogisticmodelwithv = 0.tTimesinceinoculation(d)$.

`loapproxfun` This package also provides a model-free alternative for summarising experimentally observed growth curves. The `loapproxfun` function 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.

5.3 Fitting logistic model to growth curves

These functions carry out parameter inference for the logistic model description of the observed growth curves. Inference can currently be carried out by maximum likelihood (fast, only provides point estimates for parameter values), however we are developing a set of Bayesian MCMC Gibbs sampler methods for inference which will provide distributed point estimates, making more use of the available observations. Logistic model parameter values can be used to construct quantitative fitness definitions for subsequent analysis.

5.4 Inferring genetic interaction strengths

Addinall et al., 2011 present methods for statistical epistasiis analysis based on linear error models. For appropriately designed experiments, these functions can be used to compare sets of query mutation observations with expected observations, given observations of control mutation fitnesses and the expected effect of the query mutation, given genome wide observations. Effectively, we use genome-wide observations to construct a linear predictor of query mutation fitness given control mutation fitness, and search for deviations from this prediction. Mutation fitnesses come from multiple observations and these can be summarised by mean or median fitness, and significance of deviations can correspondingly be estimated by Student's t-test or the Mann-Whitney test, both corrected for multiple comparisons. Analysis based on mean/t-test is preferred to that using median/Mann-Whitney test, since the latter have greater statistical power, however, in the case where it has not been possible to perform adequate quality control on the source data (e.g. there are occasionaly contaminants, or missing cultures, resulting in statistical outliers) the former may be preferable.

5.5 Auxiliary functions

Together with the functions for carrying out the raw analysis above, we provide several functions for visualising the data, the fit of the logistic model to the data and the visualisation of evidence for epistatic interaction. These visualisation tools are important for tracking bugs and increasing user confidence in the validity of the sophisticated QFA workflows.