Package 'lymphclon'

February 20, 2015

Version 1.3.0

Date 2014-11-06

Title Accurate Estimation of Clonal Coincidences and Abundances from Biological Replicates

Maintainer Yi Liu <liuyipei@gmail.com>

Depends R (>= 2.15.0)

Imports MASS, expm, corpcor

Description We provide a clonality score estimator that takes full advantage of the multi-biological-replicate structure of modern sequencing experiments; it specifically takes into account the reality that, typically, the clonal coverage is well below 0.1%.

License LGPL-2

Author Yi Liu [aut, cre],
Richard A. Olshen [aut],
Andrew Z. Fire [ctb],
Scott D. Boyd [ctb]

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-11 10:52:02

R topics documented:

`lymphclon-package` *Estimates the clonality score from replicate of abundances data*

Description

There are an enormous number of clones(species) distributed in a highly unequal distribution. The experimenters collect a small number of very coarse samplings, and perform noisy measurements on the samplings, leading to abundance estimations for the indiviudal biological replicates. The goal of this package is to estimate the probability that two random cells belong to the same clone. Clonality estimation arises as a naturally interesting question when trying to understand the diversity of B cell populations, T cell populations, microbial populations, cancer cell subclones, and artificial biologicl libraries, and also broader ecological settings.

This package provides two primary functions; one computes the clonality score estimate (probability that two random individuals belong to the same clone), given replicate of abundances; it also infers the underlying clonal distribution, which may be of interest in diagnostic and scientific settings.

The other function generates reasonable simulation data, for evaluation purposes. The default behavior of this function generates a very large number of classes, which is suitable for B and T cell settings. It can be easily repurposed for other applications. The simulation assumes random sampling, amplifying, and sequencing of individual cells; this is implemented by introducing sparse sampling of individual cells, followed by introduction of log-normal error, and finally unbiased "rounding" by use of a poisson distribution.

Details

Package:	lymphclon
Type:	Package
Version:	1.3.0
Date:	2014-11-06
License:	LGPL-2

Author(s)

Author and Maintainer: Yi Liu <liu.yi.pei@gmail.com>

References

Accurate Estimation of Clonal Coincidences and Abundances from Biological Replicates: lymphclon. Manuscript submitted.

Examples

```
my.data <- generate.clonal.data(n=5e3)
```

```
# n ~ 2e7 is more appropriate for a realistic B cell repertoire
my.lymphclon.results <- infer.clonality(my.data$read.count.matrix)
# a consistently improved estimate of clonality (the squared
# 2-norm of the underlying multinomial distribution)
```

`generate.clonal.data` *generate.clonal.data (part of lymphclon package)*

Description

This function generates simulated data, for evaluation purposes. We start with an underlying multinomial population with entries proportional to rank^-power distribution, where power is fixed. Next, we draw multinomially from this distribution, a fixed number of cells, to generate each desired replicate. Then, this distribution is subject to log-normal error, and subsequently scaled up to the expected number of reads. To round into integers, the expected number of reads for each clone is finally pushed through a poisson process to generate integer read counts. The poisson "rounding" process is why the resulting read counts are not exactly as specified.

Usage

```
generate.clonal.data(
  n = 2e+07,
  num.cells.taken.vector = c(2000, 5000, 10000, 20000, 50000, 50000),
  read.count.per.replicate.vector = rep(20000, length(num.cells.taken.vector)),
  clonal.distribution.power = -sqrt(2),
  pcr.noise.type = 'pareto',
  pcr.pareto.location = 1,
  pcr.pareto.shape = 1,
  pcr.lognormal.meanlog = 0,
  pcr.lognormal.sdlog = 1)
```

Arguments

`n`	The true number of distinct clones in the underlying assemblage
`num.cells.taken.vector`	A vector specifying the number of cells taken in each independent biological replicate
`read.count.per.replicate.vector`	A vector of the same length as num.cells.taken.vector, specifying the number of reads generated from each biological replicate, of the same corresponding indices
`clonal.distribution.power`	The true underlying clonal multinomial distribution is proportional to (1:n)^-clonal.distribution.power
`pcr.noise.type`	A string denoting the type of PCR noise: either 'pareto' (default), or 'lognormal'. The package author Yi Liu has found anecdotally and empirically that pareto distributions model sequencing amplification bonanzas much better than lognormal distributions.

`pcr.pareto.location`
: The location parameter for the pareto distribution; matters only if the noise type is pareto.

`pcr.pareto.shape`
: The shape parameter for the pareto distribution; matters only if the noise type is pareto.

`pcr.lognormal.meanlog`
: The meanlog parameter for the lognormal distribution; matters only if the nosie type is lognormal

`pcr.lognormal.sdlog`
: The sdlog parameter for the lognormal distribution; matters only if the nosie type is lognormal

Value

`read.count.matrix`
: This is a matrix of simulated counts, with rows corresponding to clones (classes, or species), and columns corresponding to biological replicates

`true.clone.prob`
: This is the underlying simulated assemblage multinomial distribution used to generate read.count.matrix

`true.clonality`
: This is the true clonality score of the underlying simulated assemblage

Author(s)

Yi Liu (liuyipei@stanford.edu / liu.yi.pei@gmail.com)

Examples

```
my.data <- generate.clonal.data(n=2e3)
# n ~ 2e7 is more appropriate for a realistic B cell repertoire
my.lymphclon.results <- infer.clonality(my.data$read.count.matrix)
# a consistently improved estimate of clonality (the squared
# 2-norm of the underlying multinomial distribution)
```

`infer.clonality` *infer.clonality (part of lymphclon package)*

Description

Clonality score, a useful metric used in immunology, refers to the probability that two random lymphocyte receptor chain reads drawn with replacement (makes no difference in the immunology context) from an individual corresponded to the same clone, within some given repertoire of either B cells or T cells. This package implements ane estimator which understands the multi-replicate-with-PCR structure of thes sequencing experiments.

Usage

```
infer.clonality(
  read.count.matrix,
  variance.method = 'fpc.max',
  estimate.abundances = F,
  num.iterations = 1,
  internal.parameters = list())
```

Arguments

`read.count.matrix`
: A matrix of read frequences, where each row corresponds to a distinct clone, and each column corresponds to a particular biological (rather than technical) replicate. All biological replicates should be drawn from the same person. Reads from technical replicates of the same underlying templates should be merged into a single column.

`variance.method`
: The method is a code defaulting to "fpc.max". This code determines how the covariance between the replicates – the replicates – an n by n matrix, is estimated. Other possible codes are "fpc.add", "mle.cov", and "corpcor". Empirically, from simulations ,"fpc.max" performs the best. "mle.cov" uses the empirical covariance between the replicates directly. "fpc.max" and "fpc.add" recognize that the individual off diagonal entries of this matrix should all be the same, and that they can be estimated with a simple unbiased clonality score estimate (we use the simple estimate here), but the diagonal entries differ: "fpc.max". The diagonal entries should be larger than the off diagonal entries. When the empirical variance of a replicate is lower than the off diagonal entries, "fpc.max" uses the diagonal entries' value; in contrast, "fpc.add" penalizes the replicate by adding the difference onto the off diagonal entries' values. Empirically, "fpc.max" performs best, but the margin from "fpc.add" is very very small. "corpcor" refers to a very strongly regularized method of covariance estimation from the corpcor package. This is not recommended because the shrinkage is so strong that the useful signal is lost, and performance falls to levels produced by the simple clonality estimate.

`estimate.abundances`
: A boolean value defaulting to false. If set to true, then the return value will be in the form of a list, and include an additional item named "estiamted.abundances". The estimated abundances are computed as the (conditional) precision weighted average of the read.count.matrix.

`num.iterations`
: An integer specifying the number of iterations in estimation; applicable only if the variance method is set to one of the loo or mle methods. Note that the ue.zr.half setting provides the most consistent improvement against the naive simple clonality score, in terms of MSE reduction.

`internal.parameters`
: A named list of internal parameters used primarily to speed up evaluation of MSE performance across a large number of simulations. Another use is to provide user-specified covariances internally into the method, for debugging and

evaluation purposes; the typical user should ignore this parameter. The entries are named "replicates", "rep.grahm.matrix", "simple.precision.clonality", and "use.squared.err.est". "replicates" refer to a column normalized (to 1) matrix of read.count.matrix. "rep.grahm.matrix" refers to t(replicates) "simple.precision.clonality" refers to the unweighted averaging of all n-choose-2 estimators of clonality. "use.squared.err.est" refers to a user-provided set of squared error estimates corresponding to each of the replicates; this parameter is active iff the variance method is set to 'usr.1'.

Value

`estimated.abundances`

If the estimate.abundances parameter was set to True, then return a probability vector indicating the fractional contributions of each individual clone to the overall multinomial repertoire.

`d1jkn.covariance`

Lymphclon performs a number of regularization schemes to the between comparison positive definite covariance matrix, with n-choose-2 rows and columns, where n is the number of replicates. These regularizatino schemes are jackknifed to determine their variances and covariances, so that they can be averaged. This is the covariance matrix, as determined by a leave-one-replicate out jackknife. This is return value is populated only if there are at least 4 replicates.

`estimated.squared.errs`

A numerical measure of the estimated squared 2-norm error of each given replicate.

`estimated.precisions`

A numerical measure of the estimated precisions of each given replicate.

`variance.method`

The method code provided in the input, determining the method by which the replicate-level covariances are computed.

`fpc.iter.estimates`

Specifies the intermediate clonality score estimates across iterations, when num.iterations is greater than 1. Note that, based on empirical simulations, it is best to use a num.iterations value of 1. Additional iterations occasionally, but usually do not help in improving the mean squared error of the clonality measures.

`simple.precision.clonality`

This is a base line estimator of the clonality score based on simple modeling. Briefly, for each pairing of reads possible, where the two reads arise from different biological replicates – this pairing is considered as an observation from a single share undelrying bernoulli distribution with parameter equal to the clonality score. This estimator computes maximum likelihood estimate of the underlying clonality score. This estimator can be thought of as a baseline estimate. Unlike the Gini-Simpson-Estimator, this baseline does not suffer from any convexity-based bias.

`regularized.estimates`

Under a variety of regularization settings on the n-choose-2 by n-choose-2 covariance matrix for the n-choose-2 individually unbiased clonality estimators, return the respectively weighted clonality measures for each regularization setting.

`mixture.estimates`
: There are several regularization schemes possible on the n-choose-2 by n-choose-2 covariance matrix between the n-choose pair-replicate comparisons. This estimate of clonality performs a jackknife to estimate the covariance bewteen estimates corresponding to each regularization scheme, and then performs a MLE averaging based on the covariance and the individual estimates. This is the best estimate of the clonality score provided by Lymphclon; however, it requires at least 4 replicates. There are several ways by which the selected regularized estimates can be averaged: by their jackknife covariance matrix, by their jackknife variances (ie. diagonal covariance), and simple equal weights.

`mixture.clonality`
: This defaults to the matrix weighted mixture estimate, unless it falls outside the range of the contributing regularized estimates; in which case, this value defaults to the scalar jackknife precision weighted average.

`lymphclon.clonality`
: This is the recommended clonality return value to use. This defaults to mixture.clonality; if there are only 3 replicates, then defaults to the estimate provided by the "ue.zr.half" (diagonal elements are maintained, but off diagonals are divided by 2) regularization setting on the n-choose-2 by n-choose-2 between-comparison covariance matrix.

Author(s)

Yi Liu (liuyipei@stanford.edu / liu.yi.pei@gmail.com)

Examples

```
my.data <- generate.clonal.data(n=2e3)
# n ~ 2e7 is more appropriate for a realistic B cell repertoire
my.lymphclon.results <- infer.clonality(my.data$read.count.matrix)
# a consistently improved estimate of clonality (the squared
# 2-norm of the underlying multinomial distribution)
my.lymphclon.results$lymphclon.clonality
```

Index