Beginner's guide to **rSalvador** *1.7 for Windows and Linux users*

This guide helps you get started with rSalvador. For more advanced usage, you should consult the accompanying technical manual or use the inbuilt help facility. To get inbuilt help for a function, e.g., `newton.LD`, type the R command `?newton.LD`.

1 Installing rSalvador on the Windows Platform

- Download and install R 3.6.1.
- Install `Rtools 3.5` by downloading and executing the file `Rtools35.exe`.
- Launch R.
- Install the R package `devtools` by executing the following command from within R.

```
install.packages('devtools')
```

- Execute the following `devtools` command from within R.

```
devtools::install_github("eeeeeric/rSalvador",subdir = "rsalvador")
```

2 Installation on the Linux Platform

To begin with, install R and the two R packages (`hypergeo`, `gdata`) as described in the above section. Now, follow the following five steps:

1. Make an installation directory and copy the installation file `rSalvadorV1.7.tar` into the installation directory. For example:

   ```
   mkdir myrsalvador
   ```

2. Untar the installation file that is already in your installation directory (e.g., `~/myrsalvador`):
   `tar xvf rSalvadorV1.7.tar`

3. A sub-directory `rsalvador` is now created under the installation directory (e,g., `~/myrsalvador`), but continue to stay in the installation directory. From the installation directory, install rSalvador by executing

   ```
   R CMD INSTALL rsalvador
   ```

4. If installation is not successful, go to subdirectory `~/myrsalvador/rsalvador/src` and delete all files of the form `*.o`, `*.so` and `*.rds`. Then repeat the above steps.

3 Starting rSalvador

Each time you invoke an R session, you need to load rSalvador with the R command `library(rsalvador)`.

4 Your First rSalvador Calculations

Here we use the Demerec data to demonstrate the basic capabilities of rSalvador. To view this data set, type `demerec.data`.

- to find the maximum likelihood estimate of m:

  ```
  newton.LD(demerec.data)
  ```

- to find the 95% confidence interval for m:

  ```
  confint.LD(demerec.data)
  ```

- To view the iteration details:

  ```
  newton.LD(demerec.data,show.iter=TRUE)
  ```

- to display the log-likelihood function:

  ```
  plot.likelihood.LD(demerec.data)
  ```

5 Importing and Exporting Data

There are three ways to transfer data into rSalvador.

- creating a sequence of numbers within R:

  ```
  y=c(0, 16, 20, 2, 2, 56, 3, 361, 9)
  ```

- importing data from a text file. This data file should have just one column of numbers. It can have one or more memo lines. (Use built-in help for details.)

  ```
  y=import.text.data('example1.txt')
  ```

- Typically, you may have saved your data in an Excel spreadsheet file, like the accompanying example file `example2.xlsx`. (See built-in help for details.) To import data from that file, type

  ```
  y=import.excel.data('example2.xlsx')
  ```

- Now you can repeat the calculations in the above section by replacing `demerec.data` with the new data variable `y`.

- Occasionally it may be desirable to save your data (say in `y`) into a plain text file for future use. This can be done by

  ```
  export.text.data('mytest.txt',y)
  ```

  To read this data file back into rSalvador for analysis:

  ```
  import.text.data('mytest.txt')
  ```

6 Adjusting for Plating efficiency

We use data from experiment #16 of Luria and Delbruck as an example. This experiment has a plating efficiency of 0.4. You can view the data by typing `luria.16.data` and learn more by typing `?luria.16.data`.

- to find the maximum likelihood estimate of m:

  ```
  newton.LD.plating(luria.16.data,e=0.4)
  ```

- to find the 95% confidence interval for m:

  ```
  confint.LD.plating(luria.16.data,e=0.4)
  ```

- To view the iteration details:

  ```
  confint.LD.plating(luria.16.data,e=0.4,show.iter=TRUE)
  ```

- to view the log-likelihood function graphically:

  ```
  plot.likelihood.LD.plating(luria.16.data,e=0.4)
  ```

7 Adjusting for Relative Fitness

For the sake of illustration, assume that from a fitness assay the experimentalist learns that the relative fitness is $w = 0.21$ for the Demerec data. To get a maximum likelihood estimate of m, execute the R command

```
newton.MK(demerec.data,w=0.21)
```

The output is $\hat{m} = 25.86$. To set up a 95% CI for m, execute

```
confint.MK(demerec.data,w=0.21)
```

which yields a 95% CI for m: $(22.98, 28.81)$.

8 Comparison of mutation rates

Two mutation rates can be compared by checking whether the two 84% confidence intervals overlap.

```
confint.LD.plating(crane.data[[1]],e=0.1,alpha=0.16)/3.6e9
confint.LD.plating(crane.data[[2]],e=0.1,alpha=0.16)/3.9e9
```

The two confidence intervals for the two mutation rates overlap, and hence the difference is not significant. If terminal cell population sizes are the same in the two experiments, `compare.LD` performs a likelihood ratio test. In the Newcombe experiments, the difference in N_t between Experiments F and H are small, so a likelihood ratio test is possible.

```
compare.LD(newcombe.data[[6]],newcombe.data[[8]])
```

The likelihood ratio test statistic is $\Lambda = 0.0119$ and the p-value is 0.9130.

If final cell population sizes differ noticeably, a more flexible approach is to conduct a likelihood ratio test (LRT) that accounts for the difference in final cell population size. rSalvador provides three functions for that purpose, namely, `LRT.LD`, `LRT.LD.plating` and `LRT.MK`. All these functions require a value of R, the ratio of the two cell population sizes. The function names are intended as a mnemonic aid, e.g., `LRT.LD.plating` indicates that the underlying distribution is the usual Luria-Delbrück distribution with plating efficiency less than perfect. Take the Werngren-Hoffner data for example, Here we have plating efficiency $\epsilon = 0.4$. Immediately prior to plating, the first experiment had final cell density 2.3×10^8 per mL, and the second experiment had final cell density 0.5×10^8 per mL. Therefore, R=0.5/2.3. To compare mutation rates in these two experiments, one executes the following

```
R=0.5/2.3
LRT.LD.plating(wh.data[[1]],wh.data[[2]],R=R,e1=0.4,e2=0.4)
```

which gives $\Lambda = 15.16$ and $p = 9.86 \times 10^{-5}$.

9 Accounting for excessive variation in N_t

When the initial inoculum size N_0 is too small (< 10, say), the final cell count N_t can vary considerably. If N_0 cannot be increased, the gamma mixture model can be used to reduce the potential bias caused by large variation in N_t. An estimate of the coefficient of variation (cv) for N_t is needed. For example, if the cv for N_t in the Demerec experiment were (hypothetically) 0.2, one can proceed as follows.

```
newton.B0(demerec.data, cv=0.2)
```

which produces an ML estimate of 10.96 for the parameter m. Similarly, a 95% likelihood ratio CI can be obtained by

```
confint.B0(demerec.data, cv=0.2)
```

which yields $(8.70, 13.41)$.

10 Estimating m with uncertainty about relative fitness w

The relative fitness parameter w is often measured by a competition/fitness assay in the laboratory. When the sample size is relatively large, the fundamental parameter m can be estimated while w is treated as an unknown parameter of secondary interest. For example, to find the joint ML estimate of m and w for the Demerec data, one executes:

```
newton.joint.MK(demerec.data)
```

The output is a pair of numbers, $(9.852, 1.12)$, representing an ML estimate of (m, w).

To find a 95% profile likelihood confidence interval for m, one issues the command

```
confint.profile.m(demerec.data)
```

which yields an interval of the form $(6.98, 13.01)$. The corresponding command for finding a 95% profile CI for the relative fitness w is

```
confint.profile.w(demerec.data)
```

which yields an interval of the form $(0.887, 1.447)$.

11 Determining sample size

We can use the expected Fisher information for m to determine sample size. The expected Fisher information is computed by truncating an infinite series that defines the expected Fisher information. The sample size is chosen by a ψ score that indicates the quality of the resulting 95% confidence interval for m. Mathematically, the ψ score is defined by

$$\psi = \frac{\text{half width of a 95\% CI for } m}{\text{anticipated magnitude of } m}.$$

By default the expected Fisher information for m is computed by including 3000 terms.

For example, if the experimentalist believes that $m \approx 4$ and $w \approx 0.75$. By choosing $\psi = 0.25$, one computes the required sample size by executing

`samp.size.MK(m=4,w=0.75,psi=0.25)`, which yields a sample size of 31. On the other hand, if 10% of each culture is to be plated and m is believed to be 50, then

`samp.size.LD.plating(m=50,e=0.1,psi=0.25)` yields a sample size of 15.