

DESCRIPTION OF LIBSHUFF

Here are the steps which LIBSHUFF takes in order to determine if two clone libraries are significantly different:

1. Asks the user the following questions:
 - "How many sequences are in X?"
 - "How many sequences are in Y?"

These questions are used to tell the program the size of your input matrix (in the filename "sample") and the number of sequences in each data set.

EXAMPLE:

You have two sample sets, one of 75 clones and one of 60. The total number of sequences is 135. Being first in the sorted distance matrix, the data set with 75 clones is "X" and the one with 60 is "Y".

- "How many sequences are in X? 75"
- "How many sequences are in Y? 60"

WARNING:

Make sure that the file "sample" which contains the distance matrix of all of your sample:

- (a) is a square matrix
- (b) is sorted, such that all sequences in X are first followed by sequences in Y (as below).

```
- X X X X X # # # # # #
X - X X X X # # # # # #
X X - X X X # # # # # #
X X X - X X # # # # # #
X X X X - X # # # # # #
X X X X X - # # # # # #
# # # # # # - Y Y Y Y Y
# # # # # # Y - Y Y Y Y
# # # # # # Y Y - Y Y Y
# # # # # # Y Y Y - Y Y
# # # # # # Y Y Y Y - Y
# # # # # # Y Y Y Y Y -
```

- (c) does NOT have sequence names, ONLY DISTANCES

2. The program then creates two output files, named "results" and "coverages" which will appear in the same directory as the program.
3. The program first calculates homologous (X and Y) and heterologous coverages (XY and YX) for the original libraries.

4. For the first replicate, the computer must randomly distribute clones from both libraries into two new data sets X_1 and Y_1 . The program randomly picks a number and places the sample corresponding to that row/column into Y_1 .

EXAMPLE:

The total data set is 135 x 135 samples, square. The program generates a number somewhere between 0 and 134 (a total of 135 numbers). If the number is 45, then the sample in the square matrix corresponding to row 45 (or column 45, it's the same) is placed in data set Y_1 .

The program continues to generate random numbers, always checking to make sure that it hasn't already picked that number, until there are as many samples in Y_1 as there are in Y .

5. All rows/columns which were not assigned to Y_1 are assigned to X_1 .
6. The homologous coverage of X_1 is calculated.

EXAMPLE: If $X_1 = \text{rows}[45,68,1,103,134,98,97,34, \dots, n75]$

Then the homologous coverage matrix will consist of:

$\text{row}[45,68,1,103,134,98,97,34, \dots, n75] * \text{column}[45,68,1,103,134,98,97,34, \dots, n75]$

with self-identities not being considered.

7. The heterologous coverage of Y_1/X_1 is calculated (Y_1 being the columns and X_1 the rows of the matrix).
8. The difference between the homologous coverage of X_1 and the heterologous coverage of Y_1/X_1 is determined using the appropriate equation and the resulting ΔC value is recorded.
9. The program then resets itself and randomizes the starting matrix again, creating data sets X_2 and Y_2 . This process will continue until 1000 replicates are created, calculated, and the corresponding ΔC values and heterologous coverages are stored in their respective files.
10. ΔC and p-values are printed out on the screen at the end of the program. You may wish to make a note of these numbers.
11. Two outfiles are returned, "coverages" and "results".
"coverages" contains 7 columns; D, X, Y, XY, YX, diff X/XY, and diff Y/YX.

<i>D</i>	Evolutionary distance values
<i>X</i>	Homologous coverage of X
<i>Y</i>	Homologous coverage of Y
<i>XY</i>	Heterologous coverage of X by Y
<i>YX</i>	Heterologous coverage of Y by X
<i>Diff-X/XY</i>	The values of $(C_X - C_{XY})^2$
<i>Diff-Y/YX</i>	The values of $(C_Y - C_{YX})^2$
"results" contains 3 columns; D, 95%XY, and 95%YX	
<i>D -</i>	Evolutionary distance values
<i>95%XY</i>	The values of $(C_X - C_{XY})^2$ for the 950th highest Replicate of the random shufflings (i.e. $p = 0.05$)
<i>95%YX</i>	The values of $(C_Y - C_{YX})^2$ for the 950th highest Replicate of the random shufflings (i.e. $p = 0.05$)

12. Please note, all runs of the program use the same filenames. If running multiple analyses, save your results as different names!

Also, the program has very few (if any) failsafes. An error in constructing the distance matrix or inputting the sample sizes will not be detected and the program will still run.