LACK
Statistical Analysis of Lexical Bias in Microarray Datasets
Version 3.1
Jan 24, 2005

cckim47@gmail.com

Summary

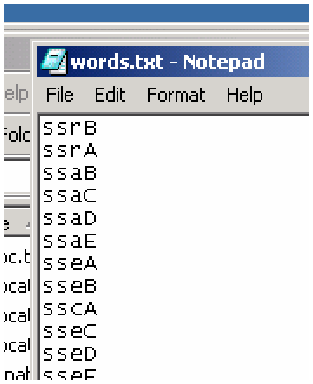
I often do a SAM or Cluster analysis and see a common theme in some of the significant genes. However, I'm usually not sure if it's really overrepresented, or just a product of observer bias. This program addresses whether or not a theme is actually overrepresented in your significant genes list. The program takes a list of significant genes and a list of user-specified search terms, and counts the number of genes which contain one of the search terms. Then, the program takes a random set of genes of the same size as the significant set from a genome annotation file and counts the hits. This process is repeated a user-specified number of times, and the counts are output in a text file.

The statistical output can either be binomial or Poisson. Because the population is known (the full dataset), the distribution can be precisely calculated using a binomial model. The output reports the precise binomial *p* values, along with the theoretical hit counts for a data set of the identical size as the significant genes list (convenient for histogram generation).

However, depending on your data set and word list sizes, the necessity to calculate large factorials may exceed the computational abilities of your computer. This will usually manifest itself as an error in the histogram values of the output file. In this case, you will need to use Poisson estimates to obtain *p*. Poisson cumulative distribution and probability mass values for the number of hits counted in the actual dataset are included in the output. The counts are tallied for convenient generation of a histogram, and raw output is also an option for flexibility in calculating different statistics.

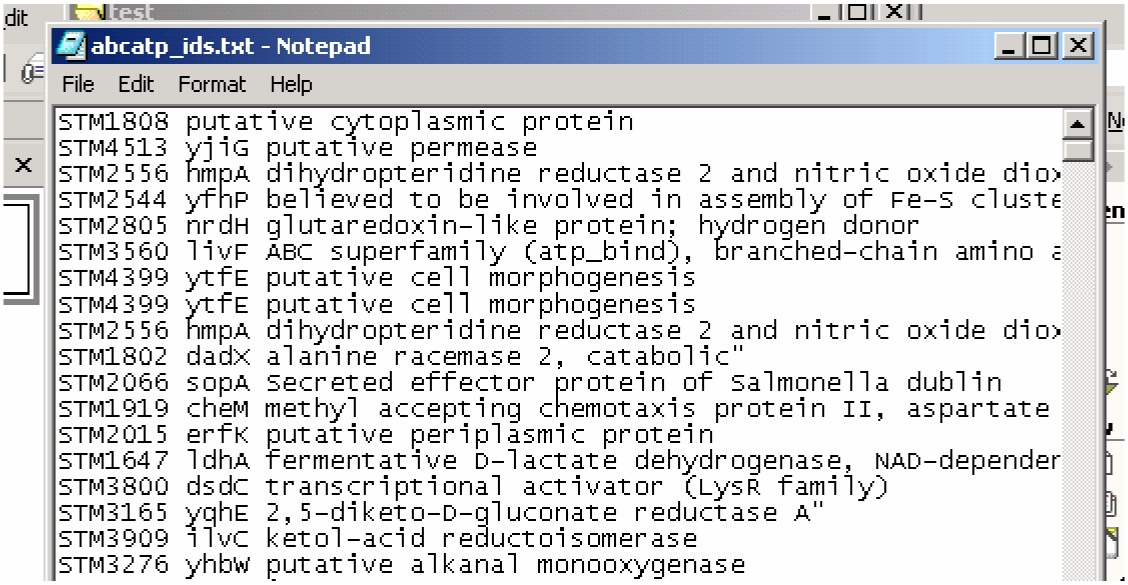
File Formats

The wordlist should be a line-delimited list saved as a text file, which you can create in programs such as Word, Excel, or Notepad.



```
ssrB
ssrA
ssaB
ssaC
ssaD
ssaE
sseA
sseB
sscA
sseC
sseD
sseE
```

Both the microarray dataset to be analyzed for lexical bias and the annotation list should be a **single** column of gene names, separated by newline characters.



```
STM1808 putative cytoplasmic protein
STM4513 yjiG putative permease
STM2556 hmpA dihydropteridine reductase 2 and nitric oxide dio
STM2544 yfhP believed to be involved in assembly of Fe-S cluste
STM2805 nrdH glutaredoxin-like protein; hydrogen donor
STM3560 livF ABC superfamily (atp_bind), branched-chain amino a
STM4399 ytfE putative cell morphogenesis
STM4399 ytfE putative cell morphogenesis
STM2556 hmpA dihydropteridine reductase 2 and nitric oxide dio
STM1802 dadX alanine racemase 2, catabolic"
STM2066 sopA Secreted effector protein of Salmonella dublin
STM1919 cheM methyl accepting chemotaxis protein II, aspartate
STM2015 erfK putative periplasmic protein
STM1647 ldhA fermentative D-lactate dehydrogenase, NAD-depender
STM3800 dsdC transcriptional activator (LysR family)
STM3165 yqhE 2,5-diketo-D-gluconate reductase A"
STM3909 ilvC ketol-acid reductoisomerase
STM3276 yhbW putative alkanal monooxygenase
```

Known Limitations and Considerations

1) Arrays are often not representative of a genome; i.e. some spots may be represented multiple times on a single array. There will therefore be a bias in the microarray output toward genes which are overrepresented on the array. It is therefore not appropriate to use a genome annotation for the annotation file. It is more accurate to use the full, filtered but pre-analysis dataset for the annotation file.

2) Choosing search terms based on the microarray dataset output can lead to user-introduced bias. For example, rare terms chosen from the dataset could lead to a false overrepresentation of the term. For example, rather than choosing terms such as “ferricytochrome” or “haeme” which would be rare, stick to general terms such as “iron,” “sulfur,” “cytochrome,” and “heme.”

3) Versions 1.2 and later use boundary-specific searching. Thus, ATP will be found in “ATP-binding protein”, but not in “CATPURR”. A log window has been added in later versions to assist with understanding the word matching algorithm.

4) If you choose to use Poisson, be sure to use enough iterations to generate a proper distribution. With the computing power available today, there is really no reason to use less than 1000 iterations; I think you could get away with 100, and more than 1000 is probably not necessary.

5) As I continue to refine the software, the command-line version is becoming obsolete. It has not been updated since version 2.02, which means that binomial statistics have not been implemented. I primarily used the command-line version for accurate benchmarking, and do not plan to support it. However, if you would like for me to update it for whatever reason, please contact me and I will consider the request.

Availability and Copyright

The program is available as a Windows executable at http://falkow.stanford.edu/whatwedo/software. The source is also available at the same website, and is written in Perl. The script requires Tk and Statistics-Descriptive. The code is copyrighted under the terms of the GNU general public license, which can be viewed at http://www.gnu.org/licenses/gpl.txt.

Version History

V3.1 completed Jan 24, 2005

- Implemented direct copy & paste from log window

V3.0 completed Sept 7, 2004

- Fixed bug in calculating cumulative p distribution which resulted in underestimation of significance
- Implemented support for high precision calculations (Math::BigInt and Math::BigFloat). Users doing significance calculations with very low p-values (<0.00001) were experiencing inaccurate results due to rounding errors. Output should now be precise to many digits, although output is now truncated at 10 digits.

V2.14 completed April 27, 2004

- Fixed bug: hangs when empty lines present in wordlist

V2.13 completed March 22, 2003

- Added cross-platform compatibility (eliminated System colors calls which were Windows-specific, modified file opening parameters, support for different newline characters)

V2.12 completed March 22, 2003

- Added built-in help

V2.11 completed March 21, 2003

- Revised GUI to include save/non-save options
- Revised GUI to include textbox clearing between runs
- Revised saving of files to separate Poisson raw output from statistics
- Revised progress bar in binomial to speed up analysis (several-fold)

V2.09 completed November 5, 2002

- Added default binomial capability, Poisson is optional

V2.07 completed October 27, 2002

- Added reporting to command line version
- Added log window to report matching (should assist user in refining search patterns)
- Added balloon help over file types
- Added program icon

V2.02 completed October 21, 2002

- Complete overhaul of algorithm – lower memory requirements, faster running time
- Removed replacement in random dataset generation

- Added command-line options (primarily for benchmarking purposes)

V1.5 completed October 14, 2002

- Removed normal statistics, implemented Poisson

V1.4 completed October 11, 2002

- Added auto-tallying of counts for histogram
- Made raw counts output optional

V1.3 completed July 17, 2002

- Added z-statistics and descriptive statistics with the Statistics-Descriptive and Statistics-Distributions modules
- Tidied output file
- Added progress bar and non-locking interface (allows exit during runtime)

v1.2 completed May 2, 2002

- Fixes bug in non-refreshing wordlist when run multiple times without closing the program. Many thanks to Scotty Merrell for pointing this out.
- Improved search algorithm – changed to case-insensitive matching, with consideration for word boundaries

v1.0 completed February 2, 2002