Code Manual for Analysis Algorithm for PathoChip Data

Daniel J. Arenas

1: Normalization

All analysis algorithms first normalize the signals so that, for each patient, summation of signals from all probes equals to one.

2: Each probe analyzed independently

Program: *1A_analyze_Single-probes_Independently.R*

This program searches for the probes for which the signals in the subjects were significantly higher than that of controls. It provides the option of performing, or not performing, a $\log_e$ transformation of the signal. This choice is to accommodate for a likely error-proportional-to-signal profile. A small constant, 10^{-6}, is added to the transformation to account for some probes in some individuals being possibly zero.

A one-sided t-test compares the probe signals of the individuals in the subject group versus those in the controls; the p-values are then adjusted using the Benjamin-Hochberg procedure.[1,2]

3: All probes searching for the same organism are averaged

Program: *1B_analyze_Organism-probe_averaged.R*

This program searches for the organisms for which the signals in the subjects were significantly higher than that of controls. It also provides the option of performing, or not performing, a $\log_e$ transformation of the signal.

The program checks that at least one probe in one subject for the organism had a non-zero reading; if not, the organism is automatically flagged as negative.

The signals are averaged over all probes searching for the same organism. For each organism, a one-sided t-test compares the probe-averages of the individuals in the subject group versus those in the controls; a p-value is stored for each organism. The p-values are then adjusted using the Benjamin H. procedure. Additional information given is the number of probes searching for the organism and how many of these probes were above the 95th percentile of all probes in the data. The average signal across probes and individuals for the subject and controls are also given.

4: Top-Percentile-Probe-Ratio (TPPR) agorithm

Program: *1C_analyze_TPPR.R*

4.1: Calculating the top-percentile cutoff

Using all probes, searching for all organisms, a percentile cutoff is calculated. Unless the user specifies otherwise, the default percentile cutoff is 95th percentile. This will later be used by the TPPR algorithm to compare between two groups how many probes had signals comparably higher than the background from all probes searching for all organisms. A

This algorithm step is convenient as it can be straightforwardly applied to any experimental data by doing a histogram on the experimental intensities from all probes. Using the background statistics should be an accurate since these experimental techniques simultaneously search for thousands of possible organisms, most of which are not expected to exist in any individual.

4.2: Comparing number of top-percentile probes

Fig 1 shows a diagram of the top-percentile-probe-ratio (TPPR) algorithm. First, all probes per organism are pooled. The individual average probe intensity $<I>_{individuals}$ is then compared to the cutoff to decide positivity. The ratio of positive probes from subjects and control are then subjected to a binomial-test with a variable significance (unless specified otherwise, 0.05). Please note that the type I error is not trivially synonymous to the significance of the *t*-test as there are two steps in the algorithm.

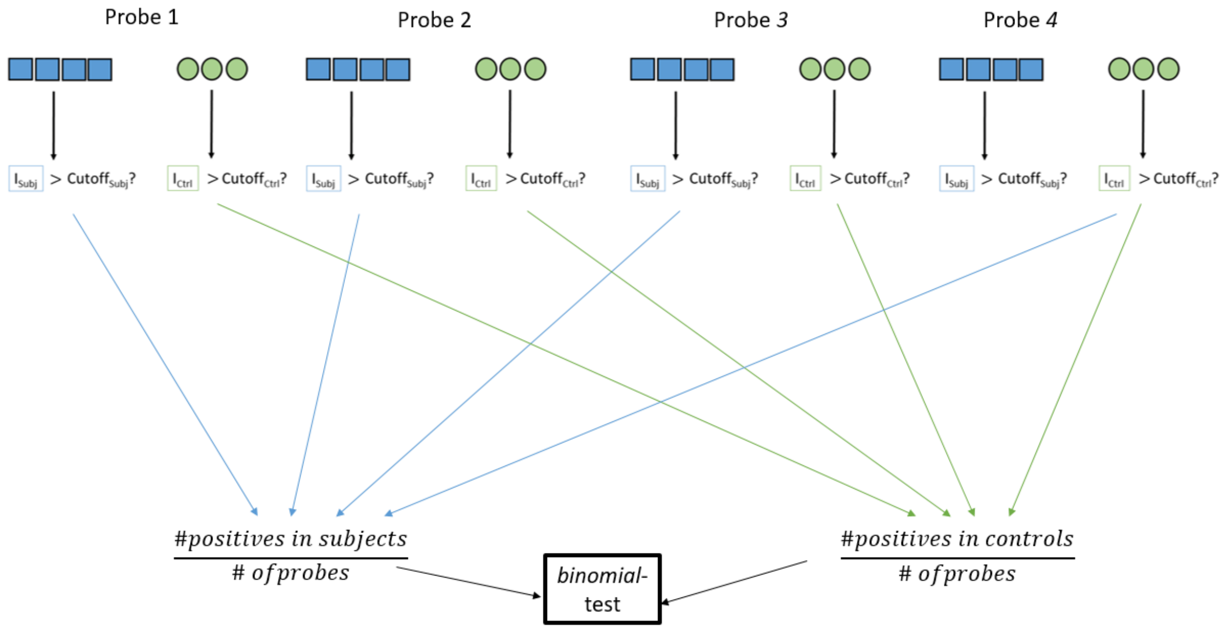

Figure 1. Diagram of TPPR algorithm to analyze multiple-probe data. All the probes for one organism are pooled. Each probe is compared to a cutoff. The number of "outlier" probes in each group are recorded. The two proportions are compared to each other by a simple binomial-test. It is important to note the intuitive result that the type I error of this algorithm is not simply the one set for the binomial-test as there is one step before comparing to a cutoff.

5: Monte Carlo simulations for estimating Type I and Type II errors along with statistical power

Program: *2_Simulate.R*

Requires functions from: https://github.com/djarenas/Simulate
Optical_Signals_from_MicroArrays

Simple mathematical formulas for the estimation of the statistical power could not be used in our analysis as the optical signals from the probes are not expected to be normally distributed. Deviation from normality also suggests that the type I error will not be trivially equal to the desired significance used in the t-test formula. The optical signals cannot be normally (or even symmetrically) distributed as the signals cannot be negative; furthermore, the sample sizes used in our iMCD/UCD study were too small to invoke the central limit theorem.

Here we used Monte Carlo simulations as they have a long useful history in estimating statistical power and type I error using both non-normal analytical distributions and experimental distributions.[3–8] The program allows the user to choose whether to simulate the data analytically, based on the theoretical description in [7], or by bootstrapping experimental data. In the analytical option, the user can choose that there is no signal above background for both subjects and controls to estimate type I error. The T1E can be calculated for any of the previous analysis algorithms as a function of sample size and number of probes searching for the same organism. The chooser can also choose a certain amount of signal above background for the subjects and calculate statistical power as a function of sample size and number of probes searching for the same organism.

In the bootstrap option, the user provides experimental signal. For example, one can use HHV-8 virus data from the HHV-8-associated-MCD lymph node sample. Using the experimental distribution from these 235 probes, the bootstrap generates a signal for a desired number of probes averaged over a desired number of subjects. And as example of a negative control, one can use the HHV-8 virus data from the remaining 18 samples consisting of UCD, iMCD, PTLD, EBV-PTLD, and reactive lymph nodes. For estimating cutoffs, the program bootstraps from all probes from all organisms to find percentile cutoffs (default of 95th percentile).

References

1. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B Methodol*. 1995;57(1):289-300.

2. Benjamini Y, Yekutieli D. The control of the false discovery rate in multiple testing under dependency. *Ann Stat*. Published online 2001:1165-1188.

3. Sánchez-Meca J, Marín-Martínez F. Homogeneity tests in meta-analysis: A Monte Carlo comparison of statistical power and Type I error. *Qual Quant*. 1997;31(4):385-399.

4. Muthén LK, Muthén BO. How to use a Monte Carlo study to decide on sample size and determine power. *Struct Equ Model*. 2002;9(4):599-620.

5. Zhang Z. Monte Carlo based statistical power analysis for mediation models: Methods and software. *Behav Res Methods*. 2014;46(4):1184-1198.

6. Kleinman K, Huang SS. Calculating power by bootstrap, with an application to cluster-randomized trials. *EGEMs*. 2016;4(1).

7. Schoemann AM, Boulton AJ, Short SD. Determining power and sample size for simple and complex mediation models. *Soc Psychol Personal Sci*. 2017;8(4):379-386.

8. Alden III RW, Hall Jr LW. Bootstrap simulations to estimate relationships between Type I error, power, effect size, and appropriate sample numbers for bioassessments of aquatic ecosystems. *J Environ Sci Health Part A*. 2020;55(13):1484-1503.