BIOINFORMATICS

Vol. 00 no. 00 2016
Pages 1–3

Combining Dependent P-values with an Empirical Adaptation of Brown's Method - Supplemental Information

William Poole [1], David L. Gibbs [1], Ilya Shmulevich [1], Brady Bernard [1*] and Theo A. Knijnenburg [1*]

[1]Institute for Systems Biology, Seattle, WA, US

1 MATHEMATICAL EXPLANATION

Here we give a more detailed mathematical explanation of the Empirical Brown's Method. We begin by explaining Brown's Method (Brown, 1975) in more detail largely following Kost and McDermott (2002). Consider k normally distributed random variables with means 0 and covariance matrix Σ,

$$X = N(0, \Sigma), \tag{1}$$

where $N(0, \Sigma)$ is an k-dimensional normal distribution. P-values can be derived from X with with a cumulative distribution function,

$$P_i = 1 - \Phi(X_i), \tag{2}$$

where P_i denotes the i^{th} P-value, X_i denotes the i^{th} component of X, and Φ denotes the cumulative distribution function. Note that this follows because the marginals of a multivariate normal distribution do not depend on the covariance. We now consider the distribution of

$$\Psi = \sum_{i=1}^{k} -2 \log P_i, \tag{3}$$

which we assume is proportional to a χ^2 distribution with $2f$ degrees of freedom, $\Psi \sim c\chi^2_{2f}$. Brown showed that

$$f = \frac{\mathrm{E}[\Psi]^2}{\mathrm{var}[\Psi]} \tag{4}$$

and

$$c = \frac{\mathrm{var}[\Psi]}{2\mathrm{E}[\Psi]} = \frac{k}{f}. \tag{5}$$

Assuming a χ^2 distribution, $\mathrm{E}[\Psi] = 2k$. Furthermore, define a new random variable $W_i = -2 \log P_i = -2 \log(1 - \Phi(X_i))$. Brown showed that,

$$\mathrm{var}[\Psi] = 4k + 2 \sum_{i<j} \mathrm{cov}(W_i, W_j). \tag{6}$$

This expression can be evaluated for each i and j via numerical integration, where

$$\mathrm{cov}(W_i, W_j) = E[W_i W_j] - 4, \tag{7}$$

$$E[W_i W_j] = \int_0^\infty \int_0^\infty w_i w_j f_{W_i, W_j}(w_i, w_j) \mathrm{d}w_i \mathrm{d}w_j, \tag{8}$$

and f_{W_i, W_j} denotes the joint distribution between W_i and W_j. Computationally, this numerical integration is slow and not suitable for large datasets (see **Supplementary Information 4**). This inspired Brown and Kost (Brown, 1975; Kost and McDermott, 2002) to find polynomial fits to calculate the covariance. While these fits work very well on low-noise normal data, we found them to be less than ideal on more realistic datasets. This led us to take a non-parametric approach and attempted to approximate $\mathrm{cov}(W_i, W_j)$ directly from the data. Let $\vec{x}_i$ be a sample drawn from X_i. We can approximate a sample, $\vec{w}_i$, from W_i by transforming the raw data using the right-sided empirical cumulative distribution function F,

$$\vec{w}_i = -2 \log(1 - F(\vec{x}_i)). \tag{9}$$

The covariance between two variables W_i and W_j can then estimated from the raw data using the well known definition of covariance,

$$\mathrm{cov}(W_i, W_j) \approx \mathrm{E}[(\vec{w}_i - \mathrm{E}[\vec{w}_i])(\vec{w}_j - \mathrm{E}[\vec{w}_j])]. \tag{10}$$

2 EBM CONVERGENCE AS A FUNCTION OF SAMPLE SIZE

We generated data from a normal distribution with $\mu = 0$ and

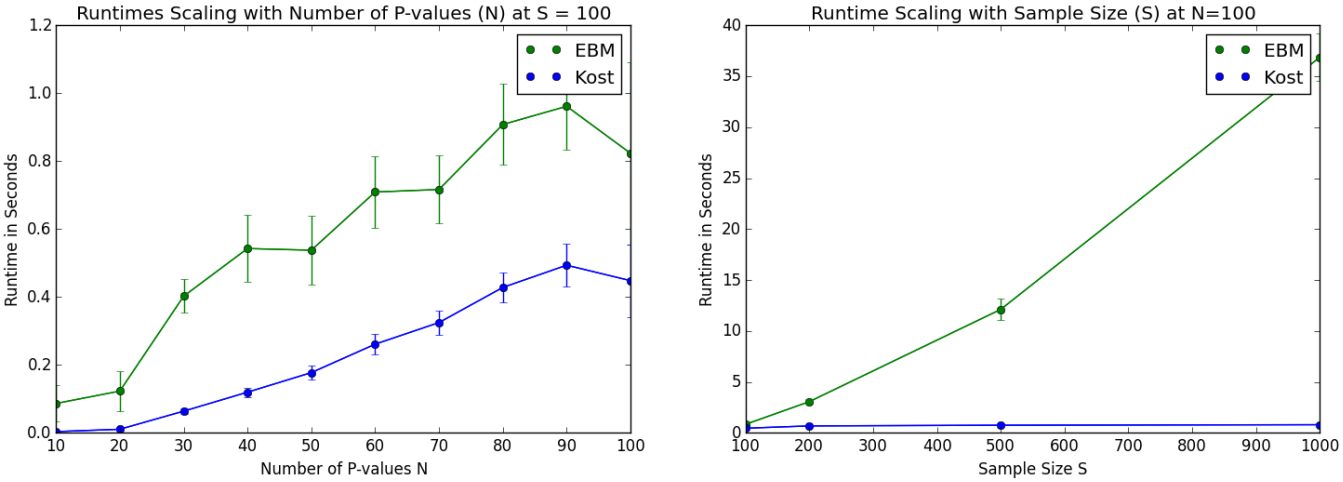
$$\Sigma = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{11}$$

No noise was added to this data, i.e. $\xi = 0$. We chose this covariance matrix because it has exactly 2 degrees of freedom (equivalent to 4 degrees of freedom for the χ^2 distribution.) We found that, given $n > 100$ samples per data vector $\vec{x}_i$, our implementation produces relatively little variation for the values of $2f$ and c (**Supp. Fig. 1**). We performed many additional tests by generating sample data from numerous other covariance matrices with known degrees of freedom. These tests demonstrated the same general convergence pattern (results not shown).

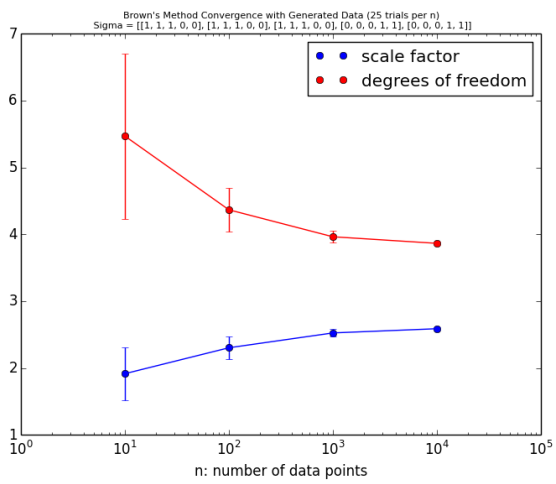
3 RUNTIMES

We note here that EBM is fast enough for use on large omics datasets. Let N be the number of P-values to be combined and S the number of samples (data points) used to generate each P-value. Our method runs in $O(N^2 S + NS \log S)$ time, where the first term comes from the pairwise covariance calculations between N P-values and the second term comes from the empirical cumulative

*Equal contribution



Supp. Fig. 2. Left: Runtime scaling with number of P-values being combined N. Sample size = 100. Error bars are the standard deviation across 50 trials from different randomly generated sets of data. Right: Runtime scaling with sample size S. $N = 100$. Error bars are the standard deviation across 50 trials from different randomly generated sets of data.



Supp. Fig. 1. ***Convergence of EBM as a function of the sample size*** n ***when calculating*** c ***and*** $2f$. A 5-by-5 covariance matrix with 2 degrees of freedom ($\Psi \sim \chi^2_4$) was used in this example. Error bars show standard deviation across 25 different trials.

distribution function calculations for each of the N random variable with S samples. For comparison, Kost's Method runs in $O(N^2S)$ time, as it doesn't use the ECDF function. Using our randomly generated data, we combined P-values for varying values of N and S (**Supp. Fig. 2**); both methods can combine hundreds of P-values based on data with thousands of samples in seconds. In comparison, direct numerical integration takes many orders of magnitude longer and is not therefore not practical for P-value combination tasks encountered in high-throughput biology data. See **Supp. Table 1** for a benchmark of running times.
Because the implementation is fast, users can empirically compute

N	S	EBM	Kost	Numerical
5	100	0.057 ± 0.027	0.0011 ± 0.0004	105.79 ± 30.01
10	100	0.086 ± 0.052	0.00347 ± 0.0013	357.05 ± 116.96
20	100	0.123 ± 0.059	0.0103 ± 0.0035	1134.24 ± 350.66
5	100	0.057 ± 0.027	0.0011 ± 0.0004	105.79 ± 30.01
5	500	0.331 ± 0.016	0.0008 ± 0.0002	73.01 ± 14.08
5	1000	1.124 ± 0.047	0.0009 ± 0.0002	74.29 ± 15.67

Supp. Table 1: ***Running Time Benchmarks*** Times shown are in seconds averaged over 50 trials generated from different sets of random data using the values of N and S indicated on the left. Numbers after the $\pm$ indicate standard deviations across the trials.

confidence intervals of the combined P-values using a bootstrap procedure if interested.

4 SOFTWARE IMPLEMENTATION

The implementation of EBM in Python uses the scipy, numpy, and statsmodels libraries. Specifically, numpy's covariance function (numpy.cov) is used to calculate the covariance and statsmodels' ECDF function is used to calculate the empirical cumulative distribution (statsmodels.distributions.empirical distribution.ECDF). This implementation is efficient and is easily applicable to large scale genomics data. Below, we describe in some detail the various components that are part of the implementation.

- Let there be k data vectors denoted $\vec{X}_1 ... \vec{X}_k$ each with n samples. Our function takes as input a matrix of these data vectors and a vector of k P-values, denoted $P_1 ... P_k$, to be combined.
- Z-Transform of the data - mean of 0 and unit variance - $\vec{Z}_i = (\vec{X}_i - \mathrm{E}[\vec{X}_i])/\mathrm{var}(\vec{X}_i)$.
- Calculate the empirical cumulative distribution function (F) over the data using the statsmodel package.

- Approximate the -2 log cumulative distribution vector, for each data vector; $\vec{w}_i = -2\log(1 - F(Z_i))$.
- For each pair of indices (i, j) calculate the covariance $\text{cov}(\vec{w}_i, \vec{w}_j)$.
- Sum covariances to calculate var$[\Psi]$, f and c.
- Calculate the combined statistic $x = -2\sum_{i=1}^{k} \log P_i$.
- Compute a meta P-value using Brown's re-scaled distribution: $P_{\text{combined}} = 1 - \Phi(\chi^2_{2f}(x/c))$, where Φ denotes the cumulative distribution function.

Additionally, for flexibility each component of our code can be called individually. This allows for the covariance matrix to be pre-computed and Brown's Method to be applied on arbitrary subsets of the data (which is how we carried out the TCGA analysis, see main text). Finally, we have included Kost's Method and Fisher's Method within our code for increased functionality and comparisons.

An efficient implementation is also available in R and Matlab. See `https://github.com/IlyaLab/CombiningDependentPvaluesUsingEBM`. The R code is also available as a Bioconductor package at `https://www.bioconductor.org/packages/devel/bioc/html/EmpiricalBrownsMethod.html`.

REFERENCES

Brown, M. B. (1975). 400: A method for combining non-independent, one-sided tests of significance. *Biometrics*, pages 987–992.

Kost, J. T. and McDermott, M. P. (2002). Combining dependent p-values. *Statistics & Probability Letters*, **60**(2), 183–190.