FEBS Letters 587 (2013) 677–682

FEBS Letters

journal homepage: www.FEBSLetters.org

The exact probability distribution of the rank product statistics for replicated experiments

Rob Eisinga [a,*], Rainer Breitling [b], Tom Heskes [c]

[a] Department of Social Science Research Methods, Radboud University Nijmegen, Netherlands
[b] Manchester Institute of Biotechnology, University of Manchester, United Kingdom
[c] Institute for Computing and Information Sciences, Radboud University Nijmegen, Netherlands

ARTICLE INFO

Article history:
Received 11 October 2012
Revised 16 January 2013
Accepted 17 January 2013
Available online 8 February 2013

Edited by Paul Bertone

Keywords:
Rank product method
Microarray
Permutational inference
Gamma approximation
Exact inference

ABSTRACT

The rank product method is a widely accepted technique for detecting differentially regulated genes in replicated microarray experiments. To approximate the sampling distribution of the rank product statistic, the original publication proposed a permutation approach, whereas recently an alternative approximation based on the continuous gamma distribution was suggested. However, both approximations are imperfect for estimating small tail probabilities. In this paper we relate the rank product statistic to number theory and provide a derivation of its exact probability distribution and the true tail probabilities.

1. Introduction

The rank product method is a popular non-parametric technique introduced by Breitling et al. [1] for identifying differentially expressed genes using data from replicated microarray experiments. It has also been widely applied to other post-genomic datasets that generate replicated rankable scores, for example in proteomics and metabolomics [3–5]. The method entails ranking genes according to their differential expression within each replicate experiment and subsequently calculating the product of the ranks across replicates. An important next step is to compare the observed rank products to their sampling distribution under the null hypothesis that the differential expression values are identically distributed (i.e., statistically exchangeable) within each of the independent experiments. Breitling et al. [1] proposed a permutation sampling procedure to approximate this distribution, whereas Koziol [2] recently suggested an alternative approximation based on the continuous gamma distribution. The latter cautions, however, that both permutation re-sampling and the gamma approximation fail to provide accurate estimates of the extreme tail probabilities of the rank product statistics.

This note provides a combinatorial exact expression for calculating the probability mass function of the rank product statistic and the exact *P*-values based on the fundamental theorem of arithmetic. The underlying method has previously been suggested by Lehner et al. [6] in a different research area, but their expression is exact only for the restricted case that the rank product is not larger than the number of genes in the array. In this paper, we remove this restriction, making the resulting counting method generally applicable to the analysis of microarray and other data. Our numerical example shows that the exact probability mass function offers an improvement over the continuous gamma approximation, which tends to understate the evidence against the null hypothesis, and permutation. This improvement is important for the application of the rank product method in all areas of biological data analysis, as the main interest is typically directed towards the tail of the distribution, that is, the detection of "significantly changed" genes, proteins or metabolites.

2. Rank product analysis

Suppose we have differential expression data for a total of n genes from k replicated experiments, with all replicates measuring the same number of genes. The underlying distribution of the differential expression values themselves is unknown, prohibiting the calculation of the probability distribution of the raw expression data. For this reason, each measurement of the differential

* Corresponding author. Fax: +31 24 3612351.
E-mail addresses: r.eisinga@maw.ru.nl (R. Eisinga), rainer.breitling@manchester.ac.uk (R. Breitling), t.heskes@science.ru.nl (T. Heskes).

0014-5793/$36.00
http://dx.doi.org/10.1016/j.febslet.2013.01.037

expression for the ith gene in the jth replicate is replaced with its rank, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k$. The most strongly up-regulated gene in each replicate is assigned rank 1 and the most strongly down-regulated gene is assigned rank n, giving k sets of ranks, denoted r_{ij}, $1 \leqslant r_{ij} \leqslant n$. Assuming no ties, each rank occurs once and only once in each replicate.

For each gene i we have a rank tuple $\{r_{i1}, \ldots, r_{ik}\}$, and rank product analysis intends to examine those tuples where all of the ranks are sufficiently small. The individual rank scores r_{ij} can be used as a test statistic for the null hypothesis that a gene is not significantly regulated against the alternative that it is differentially expressed, yielding a P-value given by $P(R \leqslant r_{ij}) = r_{ij}/n$. Rank product analysis aims to integrate the evidence from k independent biological replicates to provide a P-value for the overall test that all k single null hypotheses are true.

In line with Fisher's [7] method, the rank product approach to combining the individual P-values is to obtain the product of the ranks for gene i over the independent replicates k, i.e., $rp_i = \prod_{j=1}^{k} r_{ij}$. The observed rank product is then compared to the sampling distribution of the rank product values under the overall null hypothesis that the expression levels are identically distributed within each of the k independent replicates. Assessing the statistical significance, or P-value, of the observed expression changes therefore relies on the ability to obtain this null distribution accurately. In the original publication, Breitling et al. [1] proposed to obtain an approximate distribution under the condition that all the null hypotheses are true by permutation re-sampling. This strategy requires a computationally demanding large number of permutations to get reliable estimates of the P-values at the tails of the distribution, that is, for the most significantly changed genes. Therefore, an analytical approach for calculating the distribution without requiring permutations was desirable. Hereafter, for notational convenience, we will drop all reference to the symbol i and consider how to make probability calculations using the gamma approximation and exact calculation.

3. Gamma approximation for rank products

In [2], Koziol argues that under the null hypothesis $r_j/(n+1)$ is approximately uniformly distributed on the interval [0,1] and he uses this argument to propose a continuous gamma distribution approximation for the log-transformed rank products, $z = -\log(rp/[n+1]^k)$.

If the P-values $r_j/(n+1)$ are uniform and continuous on the unit interval [0,1], the probability distribution of $w_j = -\log(r_j/[n+1])$ is given by the exponential distribution $p(w_j) = e^{-w_j}$ with scale parameter 1, denoted as Exp(1). Given that w_j is distributed as Exp(1), the sum of w_j over k independent replicates has a gamma $(k,1)$ distribution, i.e., $p(z) = \Gamma(k)^{-1} z^{k-1} e^{-z}$, where $z = \sum_{j=1}^{k} w_j$ [see 2,6,8]. Koziol [2] shows that the empirical distribution of the log-transformed rank product values is well-approximated by the continuous gamma $(k,1)$ distribution over the (almost) entire range of support. He urges, however, that estimation of small tail probabilities of the rank products from the gamma approximation is imprecise.

The reason for the deviation is that the rank products take discrete values on the real number line, i.e., $1, 2, 3, \ldots, n^k$, whereas the continuous gamma distribution allows all non-negative real numbers. The deviations are most prominent if the rank products are small, hence at the right tail of the distribution. Below we will give an example that illustrates the difference between the true P-value and the approximate P-value based on the gamma $(k,1)$ probability density function.

4. Exact distribution of rank products

To overcome the limitations of the approximation strategies, recall that the rank products have a probability mass function. This function gives the probability that a discrete random variable RP is exactly equal to some value rp. This probability, denoted $P(RP = rp)$, can be obtained by calculating the total number of ways to get rp by multiplying k integers (number of replicates) between 1 and n (number of genes), and dividing the result by n^k. One approach to this counting problem is using a for loop. That is, run k nested loops from 1 to n, most efficiently by the divisors of rp, and count the number of times the resulting product equals rp. This brute-force search performs well, but it becomes computationally time consuming if either n or k or both are large. The more so, if in addition to the probability the P-value of large rank products is required.

An alternative calculation relies on the fundamental theorem of arithmetic also known as the unique-prime-factorization theorem [9,10]. This theorem states that every positive integer (except the number 1) has a unique prime-factorization implying that it can be presented in exactly one way as a product of powers of primes. For the problem at hand, this implies that every rank product rp greater than 1 is either prime itself or is the product of primes, i.e., $rp = p_1^{a_1} \ldots p_m^{a_m}$, where $p_1 < p_2 < \ldots p_m$ are distinct primes and the prime exponents a_t are non-negative integers. Obviously, the same goes for the divisors d of rank product rp.

We denote by $H(rp;k,n)$ the total number of representations of rank product rp as an ordered product of k ranks smaller than or equal to n. That is, two representations of rp are identical only if they contain the same ranks in the same order. We also assume by definition that $H(1;k,n) = 1$.

In their discussion of rank statistics, Lehner et al. [6] have shown that we can enumerate the number of ordered k-tuples such that their product equals rp, using

$$H(rp;k,n) = \prod_{t=1}^{m} \binom{a_t + k - 1}{k - 1} \text{ if } rp \leqslant n.$$

The computation of $H(rp;k,n)$ is an application of the so-called Piltz divisor function [11, see also Sloane's (A007425) at http://oeis.org/A007425], and intimately related to the study of ordered factorizations of integers [12]. For a proof see Nathanson [10], Theorem 7.5, and Lehner et al. [6]. The above expression for $H(rp;k,n)$ is a valid method for counting the representations of rp as long as the rank product is less than or equal to the number of genes. The function is then independent of n, and it offers the total number of ways of writing rp as an ordered product of k ranks.

This counting formula may occasionally be appropriate for examining top-lists of most up-regulated genes, if n is large and the number of replicates is small. But in many biological applications, with several replicates and noisy data, for many genes rp will be larger than n, possibly even for strongly differentially expressed genes. If that is the case, the above expression for $H(rp;k,n)$ is invalid, as it includes rank tuples with rank values that are larger than n. Obviously, such rank tuples are impossible in replicates with n genes.

Let d_g be a divisor of rp that is larger than n, where $g = 1, \ldots, v$. To obtain a generic formula that is valid for all possible rank product values, we express $H(rp;k,n)$ in terms of functions $H(\cdot;\cdot,\infty)$ as

$$H(rp;k,n) = \sum_{s=0}^{k} \sum_{\beta:\sum_g \beta_g = s} (-1)^s \binom{k}{s} \binom{s}{\beta_1, \ldots, \beta_v} \times H\left(rp/\prod_g d_g^{\beta_g}; k-s, \infty\right),$$

where β_g indicates the number of divisors equal to d_g, and β is the set of all combinations of $\beta_1,\ldots,\beta_v$, such that $\sum_g \beta_g = s$, for $s = 0,\ldots,k-1$. A proof is given in the Appendix. Notice that the function generates $H(rp;k,n)$ as a double sum over $H(\cdot;\cdot,\infty)$ which in turn is a product of combinatorial functions of prime exponents. The inner sum is taken over all combinations of divisors larger than n (with replacement) subject to the constraint that rp is dividable by the product of these s divisors, so that the remainder $rp/\prod_g d_g^{\beta_g}$ is a product of $k - s$ ranks. Hence this constraint selects sets of permissible combinations of divisors larger than n. The outer summation runs from 0 to k, but for $rp \leqslant n^{s+1}$ we only need to consider combinations of maximum s divisors. If $rp \leqslant n$, there are no divisors larger than n. In that case $s = 0$, and the expression reduces to the formula offered by Lehner et al. [6].

For example, if the rank product is not larger than n^2, hence maximum $s = 1$, we only have to consider the divisors themselves. The remainder is then always a product of $k - 1$ ranks, and the expression reduces to

$$H(rp;k,n) = \prod_{t=1}^{m}\binom{a_t + k - 1}{k-1} - \sum_{\substack{g=1\\ d_g \mid rp}}^{v}\binom{k}{1} H\,(rp/d_g;\ k-1,\infty) \text{ if } rp \leqslant n^2,$$

where the summation extends over the v divisors d_g of rp that are larger than n. The formula has a simple combinatorial interpretation. It counts the number of permutations of rank tuples with inadmissible rank values, and subtracts the aggregated result from the total number of (admissible and inadmissible) representations of rp.

If the rank product value is larger than n^2, but does not exceed n^3, thus the maximum that s can take is 2, the expression requires an extra component and becomes

$$H(rp;k,n) = H_1(rp;k,n) + \sum_{\beta:\sum_g \beta_g = 2}\binom{k}{2}\binom{2}{\beta_1,\ldots,\beta_v} \times H\,(rp/\prod_g d_g^{\beta_g};k-2,\infty) \text{ if } rp \leqslant n^3,$$

where $H_1(rp;k,n)$ is the solution for $rp \leqslant n^2$ (max s=1) and the summation is over all combinations of $s = 2$ divisors (with replacement) that satisfy the constraint that the remainder $rp/\prod_g d_g^{\beta_g}$ is a product of $k - 2$ ranks. Extensions of the formula for larger rank products are straightforward.

Once $H(rp;k,n)$ has been obtained, the probability of rp is easily calculated as $P(RP = rp) = H(rp;k,n)/n^k$. The probability for observing rp or a smaller value under the null hypothesis H_0, the P-value, is given by $P(RP \leqslant rp) = (\sum_{j=1}^{rp} H(j;k,n))/n^k$, where small P-values are evidence against H_0. Note that in determining the P-value, all piecewise-defined $H(rp;k,n)$ need to be calculated, from the most significant rank tuple possible, with $rp = 1$, to the rank product value of interest. Also, assume that the rank value is constant across replicates, then the P-value of the product increases as the number of replicates declines. This illustrates the value of using multiple experiments, in that the absence of an experiment decreases the significance.

5. Numerical example

The following is a numerical example to illustrate the calculations. Suppose a gene has the following ranks in $k = 5$ replicates, $r = 3,9,5,8,9$. Hence the rank product is $rp = 9720$. To calculate $H(9720;5,n)$, note that $9720 = 2^3.3^5.5^1$ and if $rp \leqslant n$ in 5 replicates it equals

$$H(9720;5,n) = \prod_{t=1}^{m}\binom{a_t + k - 1}{k-1} = \binom{3+5-1}{5-1}\binom{5+5-1}{5-1}\binom{1+5-1}{5-1} = 22050.$$

If the number of genes in each replicate is $n = 10000$, for example, the probability is calculated as $P(RP = 9720) = 2.20 \times 10^{-16}$, and the associated P-value is $P(RP \leqslant 9720) = 6.08 \times 10^{-14}$, much smaller than could realistically be approximated accurately using a permutation approach, even with a large number of permutations.

The rank product $rp = 9720$ has a total of $d(9720) = \prod_{t=1}^{3}(a_t + 1) = 48$ divisors. If the same rank product value is observed in $k = 5$ replicates with, for example, $n = 500$ genes each, hence $n < rp \leqslant n^2$, some divisors representing possible rank values are inadmissible in the sense that they are larger than the maximum value n. The 12 divisors in question are 540, 648, 810, 972, 1080, 1215, 1620, 1944, 2430, 3240, 4860, and 9720. The algorithm then determines for each of these divisors the number of ordered 5-tuples, including the inadmissible divisor, such that their product equals 9720. The sum over all of the 12 inadmissible divisors is subsequently subtracted from 22050. For example, for $d_g = 540$, $rp/d_g = 18 = 2^1.3^2$. If we denote by b_t the prime exponents of this remainder, then there are

$$\binom{k}{1} H\,(rp/d_g;k-1,\infty) = \binom{k}{1}\prod_{t=1}^{u}\binom{b_t+k-2}{k-2} = \binom{5}{1}\binom{1+5-2}{5-2}\binom{2+5-2}{5-2} = 200$$

ordered 5-tuples that include the integer 540. If we do the same calculation for all the divisors that are larger than n and subsequently aggregate the results, the total number of 5-tuples with an inadmissible rank value turns out to be 905. Thus the correct number of ways to get a rank product of 9720, in $k = 5$ replicates with $n = 500$ genes each, equals $H(9720;5,500) = 22050 - 905 = 21145$. The exact probability of the rank product is $P(RP = 9720) = 6.77 \times 10^{-10}$, and the P-value is calculated as $P(RP \leqslant 9720) = 1.73 \times 10^{-7}$.

To examine the accuracy of the gamma distribution approximation, we assume that the same rank product value of $rp = 9720$ is obtained in $k = 3$, 5, 10 replicates of $n = 500$, 5000, and 10000 genes. Table 1 displays for each combination of these settings the exact probability $P(RP = 9720)$, the exact P-value $P(RP \leqslant 9720)$, and the gamma approximation of the P-value $\tilde{P}_\Gamma(RP \leqslant 9720)$.

The numerical results indicate that the continuous gamma approximation fails to assume the correct form in the long right

Table 1
Exact probability $P(RP = 9720)$, exact P-value $P(RP \leqslant 9720)$, and gamma distribution approximation of the P-value $\tilde{P}_\Gamma(RP \leqslant 9720)$, for $k = 3,5,10$ replicates and $n = 500$, 5000, and 10000 genes.

k	n	P(RP = 9720)	P(RP ⩽ 9720)	$\tilde{P}_\Gamma(RP \leqslant 9720)$
3	500	4.06×10^{-6}	2.63×10^{-3}	4.27×10^{-3}
	5000	5.02×10^{-9}	3.73×10^{-6}	1.18×10^{-5}
	10000	6.03×10^{-10}	4.80×10^{-7}	1.84×10^{-6}
5	500	6.77×10^{-10}	1.73×10^{-7}	3.57×10^{-6}
	5000	7.05×10^{-15}	1.94×10^{-12}	1.82×10^{-10}
	10000	2.20×10^{-16}	6.08×10^{-14}	8.36×10^{-12}
10	500	4.50×10^{-21}	4.52×10^{-19}	1.06×10^{-13}
	5000	4.51×10^{-31}	4.59×10^{-29}	9.05×10^{-20}
	10000	4.40×10^{-34}	4.48×10^{-32}	2.34×10^{-22}

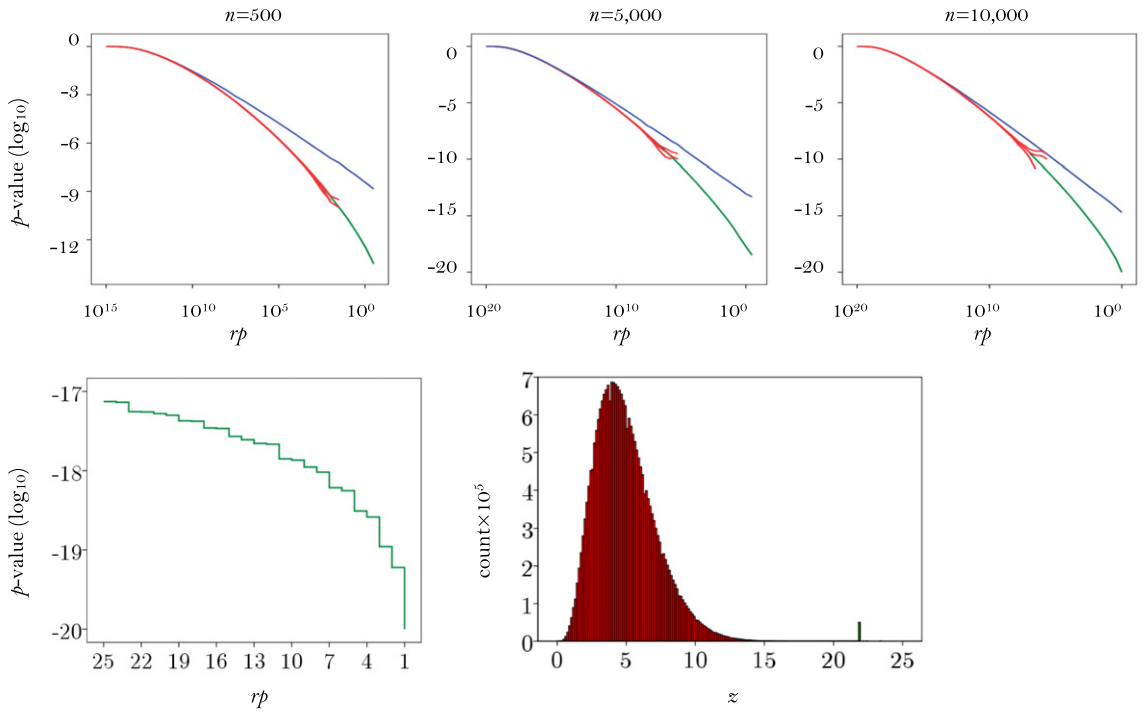

Fig. 1. *Top panel:* $\log_{10}$-transformed *P*-value for rank product of any gene, calculated by exact computation (green line), gamma (k,1) distribution approximation (blue), and permutation re-sampling (red), assuming $k = 5$ experiments and $n = 500$, 5000, and 10000 genes. The permutation model used 10^{10} random samples to approximate the distribution, where each sample consists of k randomly drawn numbers $1, \ldots, n$, for which the rank product values were calculated. The figure displays the mean and the upper and lower 95% confidence limits. *Bottom-left panel:* exact $\log_{10}$-transformed *P*-value for smallest rank products, for $k = 5$, $n = 10000$. *Bottom-right panel:* histogram of simulated distribution of log-transformed rank product statistic $z = -\log(rp/[n+1]^k)$ under the overall null hypothesis, for $k = 5$ and $n = 500$, with superimposed on the tail at $z = 21.90$ the 5-tuple {3,9,5,8,9} with rank product 9720 and exact *P*-value of 1.73×10^{-7}.

tail and that it over-predicts the true *P*-value. That is, compared to the exact *P*-value, the asymptotic gamma approximation is conservative in that it tends to understate the evidence against H_0, potentially leading to false negative results. Also, observe that the relative magnitude of the approximation error increases as the number of biological replicates rises. This implies that the approximation of small *P*-values by gamma calculation becomes increasingly unsatisfactory if the number of experiments increases.

The top panel of Fig. 1 displays the $\log_{10}$-transformed *P*-values for the entire distribution of rank products *rp* obtained by exact calculation, the approximating gamma $(k, 1)$ distribution, and by permutation re-sampling, for $k = 5$ replicates, and $n = 500$, 5000, and 10000 genes. The latter approach used 10^{10} random samples to approximate the distribution, where each sample consists of k randomly drawn numbers $1, \ldots, n$, for which the rank product values were calculated. The figure displays the mean and the upper and lower 95% confidence limits. The exact results were obtained for rank product values from 1 up to 10^7. Whereas exact probability calculation offers no computational problem unless prime factorization of very large numbers is required, calculation of the *P*-value of large rank products becomes increasingly expensive. The time needed for computing the exact *P*-values was about 200 min, using interpreted *Matlab* language running on a standard Intel Duo CPU at 2.66 GHz under Windows 7. It took approximately 100 min to obtain the permutation *P*-values with the same equipment.

As can be seen, the gamma approximation fails to perform well for $k = 5$ replicate experiments and $n = 500$ genes. Gamma calculation has considerable error for *P*-values less than 0.05, and the error increases as the *P*-values decline. Notice that the gamma approximation gains in accuracy with increasing n. Clearly, the gamma approximation has excessive error in the right tail where the rank product values are small and for exceedingly small *P*-values the approximation breaks down. But for large rank products, say 10^{10} and more, gamma calculation performs well. Indeed, for ordinary practical purposes, little seems to be lost by using the much simpler gamma approximation for rank products that are larger than 10^{10}. Gamma computations are then as good as exact. This is important because the gamma function has the obvious advantage that the required computational time is essentially negligible as compared to exact calculation and permutation.

Permutation re-sampling involves a tradeoff between accuracy and number of permutations, and thus computational time. A downside to permutation is that accurately estimating small *P*-values requires a large number of permutations. The number of permutations needed is always larger than the inverse of the *P*-value [13]. Put differently, the smallest achievable *P*-value is 1/(# permutations), but a factor of the order of 100 or so more permutations is required for reasonably accurate estimation to occur. Greater accuracy is always available, of course, but only at extra costs. This implies that the smallest *P*-values take unacceptably long amounts of time to compute. So it is (by far) not feasible to estimate them with reasonable accuracy if they need to be available on a timely enough basis.

The top panels of Fig. 1 display the results of 10^{10} permutations to accurately estimate a minimum *P*-value of 10^{-8}. The outcomes indicate that the permutation approximation is distinctly more accurate than gamma calculation, and that its accuracy is extremely good. Increasing the number of re-samples would obviously further improve the performance of permutation, but the estimation of substantially smaller *P*-values is computationally prohibitive and the smallest exact *P*-values are impossible to approximate accurately within reasonable time.

Taken together, we have a result that is of great practical value. The *P*-value of large rank products can be computed quickly by permutation (or gamma calculation), but it is unfeasible to

estimate the smallest *P*-values with permutation. In exact calculation, it is the opposite way around. The exact *P*-values of small rank products can be calculated swiftly, the smallest values even by hand. The main drawback of exact calculation is that computing the *P*-values of large rank products consumes considerable amounts of time. A substantial gain in computational time with negligible accuracy losses is possible if, for large rank products, we substitute a permutation (or gamma) estimate for the exact *P*-value. Thus, from a practical view, a quick and accurate option would be to integrate exact calculation and permutation (or gamma) approximation. Such integrated calculation method should work well with all sample and replicate sizes encountered in microarray experiments.

The bottom-left panel zooms in on the right tail with the smallest product values, i.e., $rp = 1,\ldots,25$, for $k = 5$ and $n = 10000$. The steps in the distribution of the $\log_{10}$ *P*-values are due to the discreteness of the rank product. The combinatorial calculation provides further insight into the nature of the jumps at the steps. The largest jumps in $\log_{10}$ *P*-value occur if the rank product is prime. The cumulative distribution then changes by k/n^k.

The powerful benefit of exact calculation of the *P*-values for the most significantly changed genes is shown in the bottom-right panel of Fig. 1. The figure displays a simulated distribution of the log-transformed rank product statistic $z = -\log(rp/[n+1]^k)$, under the condition that the overall null hypothesis is true, for $k = 5$ and $n = 500$, and adds to these results on the tail at $z = 21.90$ the 5-tuple {3,9,5,8,9} with rank product 9720, and an exact *P*-value of 1.73×10^{-7}. The histogram confirms the somewhat intuitive notion that it is computationally unfeasible in practice to estimate the *P*-value of most significantly changed genes with reasonable accuracy using permutation re-sampling calculation.

6. Application

To illustrate our method, expression data for bone-marrow samples from leukemia patients were obtained from Golub et al. [14], available at http://www.broadinstitute.org. The data set contains hybridizations of 27 acute lymphoblastic leukemia (ALL) and 11 acute myeloid leukemia (AML) samples on Affymetrix high-density oligonucleotide microarrays representing 7129 genes and controls.

The complete data were subjected to quantile normalization [15], and a constant was added to all measurements so that the smallest value becomes 1, following the procedure in Breitling et al. [1]. We simulated a dataset with a small number of replicates by performing a pairwise comparison of three samples of ALL and AML, similar to the analysis in Table 3 of [1], and subsequently calculated the individual ranks, the rank products and the exact *P*-values. The results are reported in Table 2. Note that the *P*-values of the highly expressed AML genes are rather small, despite the small number of replicates considered, but they are still so large that a rigorous multiple-testing correction would bring them close to the significance threshold. For example, the Zyxin gene (X95735), which an analysis of the complete dataset shows to be one of the most strongly differentiating genes between ALL and AML, has a Benjamini–Hochberg-corrected *P*-value of only 0.0056 (Bonferroni-corrected *P*-value 0.056). It is obvious that exact estimates of the *P*-values will be essential for making justified, reproducible decisions about which genes to consider as significantly differentially expressed in the downstream analysis.

7. Conclusion

In replicated microarray experiments, where typically large numbers of genes are simultaneously tested, it is crucial to be able to accurately determine small *P*-values. These values, as well as significance scores that are based on *P*-values such as the false positive rate, become all but meaningless if they are not estimated correctly. Our findings show that determining the true probability mass distribution by exact calculation offers an important improvement over the continuous gamma approximation and permutation re-sampling, at least for that part of the distribution rank product analysis is most interested in, i.e., the thin right tail.

The exact probabilities and *P*-values are easy to calculate, especially if the software program one is using performs prime factorization and produces the divisors of an integer upon command. The

Table 2
Exact *P*-values for the top 25 AML-specific genes of a subset of the leukemia data of Golub et al. [14].

Gene	Description	r_1	r_2	r_3	rp	*P*-value
M96326	Azurocidin gene (∗)	2	12	1	24	5.61×10^{-10}
L19779	Histone H2A.2 mRNA	1	3	22	66	2.30×10^{-9}
J04990	Cathepsin G precursor	46	2	2	184	9.61×10^{-9}
X17042	PRG1 proteoglycan 1, secretory granule (∗)	4	24	6	576	4.06×10^{-8}
M84526	DF D component of complement (adipsin) (∗)	14	22	9	2772	2.89×10^{-7}
M27891	CST3 cystatin C (amyloid angiopathy and cerebral hemorrhage) (∗)	130	5	7	4550	5.29×10^{-7}
U46751	Phosphotyrosine independent ligand p62 for the Lck SH2 domain mRNA (∗)	36	8	55	15 840	2.29×10^{-6}
M27783	ELA2 elastatse 2, neutrophil	87	95	3	24 795	3.79×10^{-6}
X04085	Catalase (EC 1.11.1.6) 5'flank and exon 1 mapping to chromosome 11, band p13 (and joined CDS) (∗)	17	80	21	28 560	4.45×10^{-6}
X95735	Zyxin (∗)	43	33	34	48 246	7.89×10^{-6}
M14328	ENO1 enolase 1 (alpha)	7	16	485	54 320	8.98×10^{-6}
V00594	Metallothionein isoform 2	15	13	309	60 255	1.00×10^{-5}
M20203	GB DEF = neutrophil elastase gene, exon 5	141	107	5	75 435	1.28×10^{-5}
X79234	Ribosomal protein L11	96	58	15	83 520	1.43×10^{-5}
X14008	Lysozyme gene (EC 3.2.1.17)	29	104	30	90 480	1.55×10^{-5}
X05908	ANX1 annexin I (lipocortin I)	84	11	106	97 944	1.69×10^{-5}
M63138	CTSD cathepsin D (lysosomal aspartyl protease) (∗)	26	35	109	99 190	1.71×10^{-5}
M11147	FTL ferritin, light polypeptide	16	198	36	114 048	1.99×10^{-5}
J04456	LGALS1 ubiquinol-cytochrome c reductase core protein II	22	1	6958	153 076	2.70×10^{-5}
Z19554	VIM vimentin	23	15	465	160 425	2.84×10^{-5}
U51004	Putative protein kinase C inhibitor (PKCI-1) mRNA	28	128	46	164 864	2.92×10^{-5}
X12447	ALDOA aldolase A	12	129	107	165 636	2.94×10^{-5}
X62320	GRN granulin	172	9	135	208 980	3.74×10^{-5}
M69043	Major histocompatibility complex enhancer-binding protein MAD3 (∗)	3	40	1792	215 040	3.85×10^{-5}
Y00433	GPX1 glutathione peroxidase 1	86	23	116	229 448	4.12×10^{-5}

Results for the leukemia data obtained by a pairwise comparison of three samples each from ALL and AML. Genes marked with a star (∗) are also reported amongst the top 25 AML-specific genes in the analysis of the much larger complete dataset [14].

implementation of the algorithm introduced in this note in both R and Matlab code is provided in the Supplementary data. A proof of our claim with respect to the function $H(rp;k,\ n)$ is given in Appendix A.

Acknowledgements

We are grateful to the editor and the reviewers for their constructive suggestions that enabled us to give this note a sharper focus, and we thank Ben Pelzer for implementing the algorithm in R.

Appendix A. Proof

We present an expression to evaluate $H(\rho;k,n)$, the total number of ways of writing an ordered product of ρ by multiplying k integers between 1 and n. Denote by $d_1,\ldots,d_v$ the divisors of ρ that are larger than n. We claim that

$$H(\rho;k,n) \ = \ \sum_{s\ =\ 0}^{k}\sum_{\beta:\ \sum_g \beta_g\ =\ s}(-1)^s\binom{k}{s}\binom{s}{\beta_1,\beta_2,\ldots,\beta_v} \times H\ (\rho/\prod_g d_g^{\beta_g};k-s,\infty). \tag{1}$$

Proof. Define $Q(\rho;k,\boldsymbol{\beta})$ as the number of ways to get an ordered product of ρ under the constrain that β_g of the divisors have rank d_g, for $g\ =\ 1,\ldots,v$. We are interested in $H(\rho;k,n)\ =\ Q(\rho;k,0)$, but we have an expression for $H(\rho;k,\infty)\ =\ \sum_\beta Q(\rho;k,\boldsymbol{\beta})$. Our line of reasoning is to express $H(\rho;k,\infty)$ in terms of $Q(\cdot;\cdot,0)\ =H\ (\cdot;\cdot,n)$, and then to invert this relationship.

By dividing out a single divisor d_g, we get the relationship

$$Q(\rho;k,\boldsymbol{\beta})\ =\ kQ(\rho/d_g;k-1,\beta_1,\ldots,\beta_g-1,,\ldots,\beta_v)/\beta_g,$$

and repeating this until all divisors have been divided out, we obtain

$$Q(\rho;k,\boldsymbol{\beta})=c(k,\boldsymbol{\beta})Q(f(\boldsymbol{\beta})\rho;k-s(\boldsymbol{\beta}),0),$$

$$\text{with } s(\boldsymbol{\beta})=\sum_g \boldsymbol{\beta}_g,$$

$$c(k,\boldsymbol{\beta})=\frac{k!}{\prod_g \boldsymbol{\beta}_g!(k-s(\boldsymbol{\beta}))!}=\binom{k}{s(\boldsymbol{\beta})}\binom{s(\boldsymbol{\beta})}{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_v},\ \text{and} \tag{2}$$

$$f(\boldsymbol{\beta})=\prod_g d_g^{-\boldsymbol{\beta}_g}$$

Substituting this into the definitions of $H(\rho;k,\infty)$ gives

$$H(\rho;k,\infty)\ =\ \sum_\beta Q(\rho;k,\boldsymbol{\beta})\ =\ \sum_\beta c(k,\boldsymbol{\beta})H(f(\boldsymbol{\beta})\rho;k-s(\boldsymbol{\beta}),n). \tag{3}$$

Similarity with, for example, the Möbius inversion formula and the inclusion–exclusion principle, suggests the inversion

$$H(\rho;k,n)\ =\ \sum_\beta(-1)^{s(\beta)}c(k,\boldsymbol{\beta})H(f(\boldsymbol{\beta})\rho;k-s(\boldsymbol{\beta}),\infty),$$

which is equivalent to the solution (1) above. To verify that this solution is indeed correct, we substitute it into (3):

$$H(\rho;k,\infty)=\sum_\beta\sum_\gamma c(k,\boldsymbol{\beta})(-1)^{s(\gamma)}c(k-s(\boldsymbol{\beta}),\boldsymbol{\gamma})H(f(\boldsymbol{\beta})f(\boldsymbol{\gamma})\rho;k-s(\boldsymbol{\beta})-s(\boldsymbol{\gamma}),\infty) \tag{4}$$

From the definitions (2) we have

$$f(\boldsymbol{\beta})f(\boldsymbol{\gamma})\ =\ f(\boldsymbol{\beta}\ +\ \boldsymbol{\gamma}),\ \text{and}\ c(k,\boldsymbol{\beta})c(k\ -\ s(\boldsymbol{\beta}),\boldsymbol{\gamma}) \ =\ c(k,s(\boldsymbol{\beta}\ +\ \boldsymbol{\gamma}))\prod_g\binom{\boldsymbol{\beta}_g+\boldsymbol{\gamma}_g}{\gamma_g}$$

Making a change of variables, $\boldsymbol{\delta}\ =\ \boldsymbol{\beta}+\boldsymbol{\gamma}$, and realizing that since $\beta_g\ \geqslant 0$ we have to constrain $\gamma_g\leqslant\delta_g$, (4) becomes

$$H(\rho;k,\infty)=\sum_\delta\left[\prod_g\sum_{\gamma_g\ =\ 0}^{\delta_g}(-1)^{\gamma_g}\binom{\delta_g}{\gamma_g}\right]c(k-s(\boldsymbol{\delta})),\boldsymbol{\delta})H(f(\boldsymbol{\delta})\rho;k-s(\delta),\infty).$$

Now, since

$$\sum_{\gamma_g\ =\ 0}^{\delta_g}(-1)^{\gamma_g}\binom{\delta_g}{\gamma_g}\ =\ \begin{cases}1\,\text{if}\,\delta_g\ =\ 0\\ 0\,\text{otherwise},\end{cases}$$

we see that all terms in the sum over $\boldsymbol{\delta}$ cancel, except for $\boldsymbol{\delta}=\mathbf{0}$, which indeed yields $H(\rho;k,\infty)\ =\ H(\rho;k,\infty)$, and this concludes the proof. □

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.febslet.2013.01.037.

References

[1] Breitling, R., Armengaud, P., Amtmann, A. and Herzyk, P. (2004) Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. FEBS Lett. 573, 83–92.
[2] Koziol, J.A. (2010) Comments on the rank product method for analyzing replicated experiments. FEBS Lett. 584, 941–944.
[3] Smit, S., van Breemen, M.J., Hoefsloot, H.C.J., Smilde, A.K., Aerts, J.M.F.G. and de Koster, C.G. (2007) Assessing the statistical validity of proteomics based biomarkers. Anal. Chim. Acta 592, 210–217.
[4] Wiederhold, E., Gandhi, T., Permentier, H.P., Breitling, R., Poolman, B. and Slotboom, D.J. (2009) The yeast vacuolar membrane proteome. Mol. Cell. Proteomics 8, 380–392.
[5] Fukushima, A., Kusano, M., Redestig, H., Arita, M. and Saito, K. (2011) Metabolomic correlation-network modules in Arabidopsis based on a graph-clustering approach. BMC Syst. Biol. 5, 1.
[6] Lehner, M.J., Coehlo, N.K., Zhang, Z.-W., Bianco, F.B., Wang, J.-H., Rice, J.A., Protopapas, P., Alcock, C., Axelrod, T., Byun, Y.-I., Chen, W.P., Cook, K.H., de Pater, I., Kim, D.-W., King, S.-K., Lee, T., Marshall, S.L., Schwamb, M.E., Wang, S.-Y. and Wen, C.-Y. (2010) The TAOS project: statistical analysis of multi-telescope time series data. Astr. Soc. P. 122, 959–975.
[7] Fisher, R.A. (1932) Statistical Methods for Research Workers, Oliver and Boyd, London.
[8] Balakrishnan, N. and Nevzorov, V.B. (2003) A Primer on Statistical Distributions, Wiley, Hoboken, NJ.
[9] LeVeque, W.J. (1977) Fundamentals of Number Theory, Addison-Wesley, Reading, Mass.
[10] Nathanson, M.B. (2000) Elementary Methods in Number Theory, Springer, New York.
[11] Canfield, E.R., Erdös, R. and Pomerance, C. (1983) On a problem of Oppenheim concerning "Factorisatio Numerorum". J. Number Theory 17, 1–28.
[12] Knopfelmacher, A. and Mays, M. (2006) Ordered and unordered factorization of integers. Math. J. 10, 73–89.
[13] Knijnenburg, Th.A., Wessels, L.F.A., Reinders, M.J.T. and Shmulevich, I. (2009) Fewer permutations, more accurate *P*-values. Bioinformatics 25, 161–168.
[14] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.
[15] Bolstad, B.M., Irizarry, R.A., Åstrand, M. and Speed, T.P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics 19, 185–193.