中国科学: 信息科学 2017 年 第 47 卷 第 10 期: 1334–1348
SCIENTIA SINICA Informationis
论文

磷酸化基序精确置换检验 p-value 的计算方法

吴军[1*], 段琼[2], 张琳[1], 何增有[2,3*]

1. 遵义师范学院信息工程学院, 遵义 563000
2. 大连理工大学软件工程学院, 大连 116620
3. 辽宁省泛在网络与服务软件重点实验室, 大连 116620
* 通信作者. E-mail: wujun.myway@gmail.com, zyhe@dlut.edu.cn

收稿日期: 2017–01–12; 接受日期: 2017–02–09; 网络出版日期: 2017–08–30
国家自然科学基金 (批准号: 61572094) 资助项目

摘要 蛋白质磷酸化基序指的是位于磷酸化位点周围具有位置特殊性的氨基酸序列. 磷酸化基序挖掘是生物信息学中的一个重要问题. 针对该问题, 已经提出了一些有效的挖掘方法. 但是, 这些方法所挖掘到的磷酸化基序中会存在很多的假阳性结果. 采用这些假阳性结果做进一步研究会导致严重的误导. 通常, 可以采用统计显著性检验来过滤掉这些无意义的磷酸化基序. 置换检验是统计显著性检验中一种常用的方法, 但该方法存在 3 个明显的缺点, 这些缺点降低了置换检验的实用性. 本文分析导致置换检验 3 个缺点的原因, 提出了一个叫做 EPPM 的算法来计算所有被检验的磷酸化基序的精确置换检验 p-value. 实验证明, EPPM 算法能够成功消除掉置换检验中的 3 个缺点, 并且在几个性能评估指标上要胜过非精确置换检验算法. 据作者所知, 这是目前唯一能计算精确置换检验 p-value 进行磷酸化基序结果评估的研究工作.

关键词 磷酸化基序挖掘, 基序评估, 统计显著性检验, 置换检验, 精确置换检验 p-value

1 引言

对于大多数生物过程的运转和维持而言, 蛋白质磷酸化是一种非常重要的蛋白质翻译后修饰活动. 该活动在大量细胞生命活动中都有着重要的作用, 这些生命活动包括: 新陈代谢, 信号传导, 细胞分裂、移动、变异和死亡等 [1,2]. 高通量质谱技术的应用极大程度上推动了磷酸化的研究, 比如应用串联质谱技术能够在实验中发现大规模的磷酸化位点 [3,4].

磷酸化基序指的是位于磷酸化位点上下游的氨基酸序列. 磷酸化基序挖掘的目标是从磷酸化肽段数据集 P 和非磷酸化肽段数据集 N 中找到出现频率有显著不同的基序. 这种不同是指这些基序在 P 中出现的频率要在一定程度上大于在 N 中出现的频率 [5,6]. 通常, P 集合被称为前景集合, N 集合被

引用格式: 吴军, 段琼, 张琳, 等. 磷酸化基序精确置换检验 p-value 的计算方法. 中国科学: 信息科学, 2017, 47: 1334–1348, doi: 10.1360/N112017-00012
Wu J, Duan Q, Zhang L, et al. Computing exact permutation p-values for phosphorylation motifs (in Chinese). Sci Sin Inform, 2017, 47: 1334–1348, doi: 10.1360/N112017-00012

称为背景集合. 挖掘到的这些磷酸化基序可以提供激酶活跃的信息, 也可以反映潜在的调节机制, 从而能够促进未知的磷酸化位点的预测.

鉴于磷酸化基序具有重要的生物意义, 针对磷酸化基序挖掘问题, 已经提出了一些有效的算法 [7]. 在这些方法中, Motif-X 方法 [5] 和基于 Motif-X 的方法 [8,9] 都采用了贪心算法的思想挖掘磷酸化基序; MoDL 方法 [6] 尝试从最大化磷酸化基序的表达角度出发去寻找磷酸化基序. 但以上方法都不能够保证挖掘结果的完整性. 为了解决这个问题, Motif-ALL 方法 [10] 运用了 Apriori 算法 [11] 找到所有统计显著的磷酸化基序. 尽管 Motif-ALL 方法能够保证磷酸化基序的完整性, 但在它挖掘到的磷酸化基序中会存在许多冗余, 即一些磷酸化基序的显著性是由它的子基序的显著性造成的. C-Motif 方法 [12] 运用了两个度量来对挖掘到的磷酸化基序进行评估, 从而能够去除掉许多冗余的基序, 但该方法也会去除掉一部分真的磷酸化基序. 由于上述方法都没有实施多重假设检验来判别挖掘到的磷酸化基序的真假性, 从而会导致挖掘结果中会存在一部分假阳性结果.

置换检验是统计显著性检验中一种常用的检验方法. DSP 方法 [13] 将标准置换检验直接运用到了磷酸化基序的评估中, 并通过一系列实验证明了该方法可以去除掉许多假阳性结果. 通常情况下, 置换检验通过直接枚举所有的置换数据集合来得到精确零分布是不可行的. 因此在实际应用中, 零分布通常由所有置换数据集合的一个子集来生成.

由于置换的随机性, 标准置换检验方法存在 3 个缺点. 第一, 计算出的 p-value 可能为 0; 第二, 每次执行该算法计算得出的同一磷酸化基序的 p-value 几乎都不相同. 第三, 时间代价很高. 由于上述缺点, 重复执行该置换检验算法可能会得到不同的结果. 分析发现上述 3 个缺点具有相关性, 其导致原因是不精确的零分布, 即每次执行该置换检验所得到的零分布只是精确零分布的一个近似. 因此, 由该零分布计算得出的 p-value 也是不精确的. 如果通过某种方法能够快速计算出精确 p-value, 便能够去除掉上述 3 个缺点. 通过分析磷酸化基序在置换数据集合中的产生过程, 提出了 EPPM 方法来计算磷酸化基序的精确置换检验 p-value. 该算法能够避免实际产生所有可能的置换数据集合直接计算得到精确零分布, 从而能够计算出精确置换检验 p-value. 通过实验对比 EPPM 和 DSP 方法, 体现了在磷酸化基序挖掘中计算精确 p-value 的优越性.

2 问题定义

2.1 磷酸化基序

磷酸化基序通常被表示为一段固定长度的字符串, 该字符串由磷酸化残基 (表示为 S, T 或 Y)、位置固定的氨基酸和位置不固定的氨基酸组成. 其中, 位置固定的氨基酸直接用该氨基酸表示, 位置不固定的氨基酸用 “.” 表示. 例如, 一个磷酸化基序的残基为 S, 在残基的下游第 3 个位置处有一个固定的氨基酸 G, 在上游第 1 个位置处有一个固定的氨基酸 A, 下游第 1 和 2 个位置处均是一个不固定的氨基酸, 那么该磷酸化基序对应的字符串表示为 (AS..G). 如果一个磷酸化基序 m 除了残基之外, 位置固定的氨基酸有 k 个, 那么这个基序的长度为 k.

如果一个磷酸化基序 m 是一条肽段 t 的子序列, 则称 t 包含 m, 表示为 $m \subseteq t$. m 在前景集合 P 中的支持度 $\sup(m, P)$ 被定义为 P 中包含 m 的肽段条数, 即 $\sup(m, P) = |\{t|t \in P \wedge m \subseteq t\}|$. 如果一个基序的支持度 $\sup(m, P)$ 大于等于用户定义的一个支持度阈值 min_sup, 则称该基序在 P 集合中是频繁基序.

2.2 磷酸化基序挖掘

自 Agrawal 等在数据挖掘领域中提出频繁模式挖掘问题以后 [11], 目前已有许多有效的频繁模式挖掘算法 [14,15]. FP-growth 算法 [16] 是一种常用的频繁模式挖掘算法, 该算法采用分治的策略, 将找到最长模式的问题转化成递归地寻找更短模式的问题, 然后再把它们的后缀拼接起来得到所有的频繁模式. 由于磷酸化基序挖掘的目的是找到相比背景集合 N 而言, 在前景集合 P 中更频繁出现的基序, 因此将应用 FP-growth 算法找到 P 集合中的频繁磷酸化基序; 随后再计算出这些频繁磷酸化基序的统计度量值; 最后用置换检验去除掉非统计显著的磷酸化基序.

2.3 统计显著性检验

在统计显著性检验中, 存在两种假设: 零假设和备择假设 [17]. 零假设是磷酸化基序在前景集合 P 和背景集合 N 中具有相同的分布. 每一个磷酸化基序都被赋予一个 p-value 来度量它的统计显著性. p-value 的值越小, 其统计显著性越强. 通常, 0.05 是 p-value 的一个常用阈值, 它表示一个纯粹偶然出现的磷酸化基序被错误地认为是统计显著的磷酸化基序的概率是 0.05. 在统计学中, 这种本身并不是统计显著的但被错误地认为是统计显著的磷酸化基序被称为假阳性结果.

磷酸化基序挖掘算法通常会挖掘到很多的磷酸化基序, 并且这些磷酸化基序需要被同时检验, 这样的情况称为多重假设检验. 伪发现率 (FDR) 是一种被广泛应用于多重假设检验中控制假阳性结果数量的度量. FDR 的含义是错误拒绝原假设的个数占所有被拒绝的原假设个数的比例的期望值. FDR 可以用 Benjamini 和 Hochberg 提出的方法 (简称 BH 方法) [18] 来控制.

2.4 置换检验

文献 [13] 中的 DSP 方法将标准置换检验应用到了挖掘统计显著的磷酸化基序中. 标准置换检验具有如下 3 个步骤:

第 1 步, 构建一个置换检验零假设, 并选择一个合适的统计度量.

第 2 步, 将 P 和 N 集合中的肽段进行随机置换, 随之挖掘出该置换数据集合中在 P 集合中出现次数更频繁的磷酸化基序. 反复置换多次, 用所有挖掘到的磷酸化基序的统计度量值来构建零分布.

第 3 步, 由该零分布计算出所有被检验的磷酸化基序的 p-value.

在多数情况下, 由于计算所有可能的置换数据集合的不切实际性, 零分布通常由所有置换数据集合的一个子集来构建. 因此, 标准置换检验得到的零分布只是精确零分布的一个近似分布. 研究发现非精确的零分布存在以下 3 个缺点:

(1) 如果一个磷酸化基序的统计度量值很大 (假设该值越大, 则其对应的磷酸化基序的统计显著性越强), 它的 p-value 将有很大概率为 0, 这是一个极度不好的近似值.

(2) 由于置换的随机性, 多次执行该置换检验算法会导致同一个磷酸化基序得到不同的 p-value.

(3) 标准置换检验算法的计算开销很大, 为了追求一个更精确和稳定的结果, 通常需要增加置换次数, 这将会导致更大的计算开销.

由于每次执行该算法得到的零分布都是由所有置换数据集合的一个随机子集得到的, 因此通过该零分布计算得出的 p-value 也只是一个近似值, 这是导致以上缺点的共同根本原因. 如果能够计算出精确置换检验 p-value, 便能够去除掉上述 3 个缺点. 同时, 在标准置换检验中, 如果置换的次数远远小于该集合所有可能的置换的数量, 该算法将很难保证得到的零分布是精确零分布的一个合理近似. 显

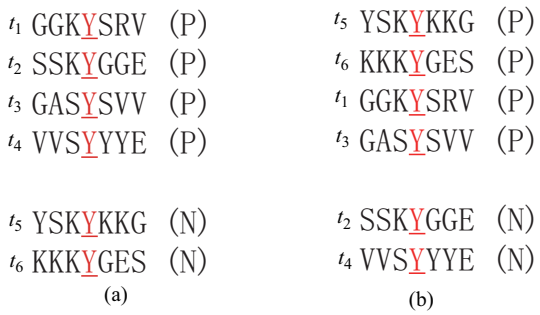


图 1 (网络版彩图) (a) 原始数据集合; (b) 置换数据集合

Figure 1 (Color online) (a) The original data set; (b) the permuted data set

然, 由一个不合理的近似零分布计算得出的一个磷酸化基序的 p-value, 将会与它精确的 p-value 产生严重偏差, 这将严重误导随后的决策.

3 EPPM 算法

在标准置换检验中, 如果想要得到磷酸化基序的精确 p-value, 需要生成所有可能的置换数据集合来建立零分布. 如前所述, 由于巨大的计算开销, 这种直接生成所有可能的置换数据集合的方法是不现实的. 为了计算被检验的磷酸化基序的精确 p-value, 借鉴文献 [19] 中的思想, 提出了一个叫做 exact permutation p-values for phosphorylation motifs (EPPM) 的算法, 该算法能够在不用实际产生所有可能的置换数据集合的情况下, 直接计算得到精确零分布.

3.1 置换过程

将 P 和 N 集合中的肽段进行了编号, 且每条肽段的编号都是唯一的. 置换过程包含如下两个步骤: 首先, 生成一个肽段编号的随机置换序列; 然后, 将该随机序列前 $|P|$ 个编号对应的肽段放入到 P 集合中, 余下的肽段放入到 N 集合中. 例如, P 集合中包含 4 条肽段 (编号为 t_1, t_2, t_3, t_4), N 集合中包含 2 条肽段 (编号为 t_5, t_6), 并且每条肽段包含 6 个氨基酸和 1 个磷酸化残基, 如图 1(a) 所示. 首先生成一个随机的肽段编号序列, 假设该序列为 t_5, t_6, t_1, t_3, t_2, t_4, 然后根据该序列将 t_5, t_6, t_1, t_3 对应的肽段放入到 P 集合中, 将 t_2, t_4 对应的肽段放入到 N 集合中, 这样就完成了一次置换, 置换结果如图 1(b) 所示. 图 1(a) 这样的集合被称为原始数据集合, 图 1(b) 这样的集合被称为置换数据集合.

3.2 统计度量

统计度量可以提供足够的信息, 从而能够做出拒绝或者接受零假设的决定. 在磷酸化基序挖掘中, 统计度量是用来度量磷酸化基序在 P 和 N 集合中是否具有相同的分布. 采用 z-value 作为磷酸化基序的统计度量. 假设一个给定的磷酸化基序 m 在 P 集合中的支持度为 s, 那么该磷酸化基序 m 的 Odds Ratio [10] 的计算公式为

$$\mathrm{OR}(m,s)=\frac{s\times(|N|-\sup(m,D)+s)}{(\sup(m,D)-s)\times(|P|-s)}, \tag{1}$$

其中 $|\cdot|$ 表示集合中肽段的数量, D 表示的是 $P \cup N$ 集合. 如果 m 的 Odds Ratio 大于 1, 则说明 m 在 P 中更容易出现. 为了进行统计推断, 对上述 Odds Ratio 取对数, 得到样本 Log Odds Ratio:

$$\mathrm{LOR}(m,s) = \log(\mathrm{OR}(m,s)). \tag{2}$$

上述 Log Odds Ratio 的标准错误等于

$$\mathrm{SE}(m,s) = \sqrt{\frac{1}{s} + \frac{1}{|N| - \sup(m,D) + s} + \frac{1}{\sup(m,D) - s} + \frac{1}{|P| - s}}. \tag{3}$$

从而, 得到 z-value 的计算公式为

$$z(m,s) = \mathrm{LOR}(m,s)/\mathrm{SE}(m,s). \tag{4}$$

从上述置换过程可知, 置换并不改变 $|P|$ 和 $|N|$ 的值, 仅仅只是改变了一个磷酸化基序 m 在 P 集合和 N 集合中的支持度大小, 即 $\sup(m,P)$ 和 $\sup(m,N)$. 由于置换过程也不涉及到肽段内部氨基酸的改变, 从而 $\sup(m,P)$ 与 $\sup(m,N)$ 之和在所有置换数据集合中都是相等的, 即 $\sup(m,D)$ 是一个定值. 因此, 一个磷酸化基序 m 的 z-value 实际上是由它在 P 集合中的支持度 s 决定的.

3.3 p-value

一个磷酸化基序 m 的 p-value 是指假设 m 在 P 和 N 集合中具有相同分布的情况下, 获得一个至少和 m 的原始统计度量值一样大的磷酸化基序的概率. 假设 $S = \{z_1, z_2, \ldots, z_n\}$ 是一个包含所有统计度量值的集合, z_q 表示磷酸化基序 m 在原始数据集合上的统计度量值, 则 m 的 p-value 计算公式为

$$p(m) = \frac{|\{z_j | z_j \geqslant z_q, z_j \in S\}|}{|S|}. \tag{5}$$

3.4 EPPM 算法

精确零分布的 x 轴是统计度量值, y 轴是这些统计度量值对应的频率. 如果所有的统计度量值和其相应的频率能够直接计算出来, 就能够避免实际生成所有可能的置换数据集合而得到精确零分布. 基于这个设想, EPPM 算法通过模拟置换数据集合的生成来直接计算所有统计度量值和其相应的频率.

由于每个置换数据集合的 P 集合中的肽段不尽相同, 从而由每个置换数据集合挖掘到的磷酸化基序也不一定相同. 计算精确零分布的首要步骤是挖掘出所有置换数据集合中可能出现的磷酸化基序. 置换方法并不改变肽段内部的氨基酸, 因此可以在整个集合 D 上运行 FP-growth 算法, 找到所有可能出现的频繁磷酸化基序.

如前所述, 统计度量值是由磷酸化基序在 P 集合中的支持度决定的, 并且一个磷酸化基序 m 在 P 集合中的每一个支持度 s 仅仅只对应于一个统计度量值. 如果能够计算出每个磷酸化基序在所有置换数据集合中的支持度范围, 那么所有的统计度量值都能够通过式 (4) 直接计算出. 对于一个给定的磷酸化基序 m, 它在 P 集合中支持度的下界 $L(m)$ 是 $\max\{0, |P| + \sup(m,D) - |D|\}$, 它在 P 集合中支持度的上界 $U(m)$ 是 $\min\{|P|, \sup(m,D)\}$. 因此, m 在 P 集合中的支持度范围是 $s \in [L(m), U(m)]$.

对于一个磷酸化基序 m 而言, 它的某个统计度量值 $z(m,s)$ 对应的是 m 在 P 集合中支持度为 s 的情景, 那么计算 $z(m,s)$ 频率的问题就能够转化为计算 m 在置换数据 P 集合中支持度为 s 的情景的

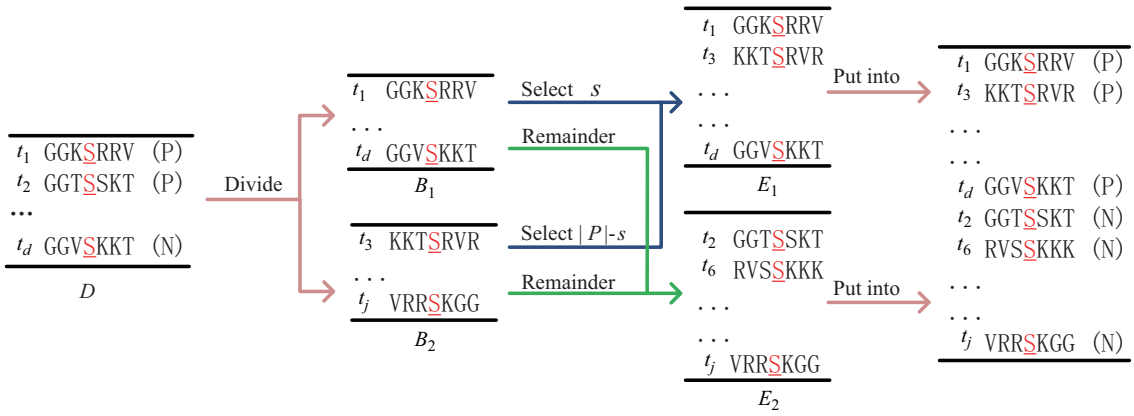


图 2 (网络版彩图) P 中仅有 s 条肽段包含磷酸化基序 m 的置换数据集合的模拟过程

Figure 2 (Color online) The generation of permuted data sets where only s peptides contain motif m in P

频率. m 在置换数据 P 集合中支持度为 s 的情景的频率, 具体指的是 P 集合中仅有 s 条肽段包含 m 的置换数据集合的数量. 通过模拟此置换数据集合的产生, 可以计算出该置换数据集合的数量. 图 2 展示了该模拟过程. 首先, 将 D 集合中的肽段分成两个集合: B_1 和 B_2, 其中 B_1 集合中的肽段都包含 m, B_2 集合中的肽段都不包含 m. 显然, B_1 和 B_2 集合的大小分别是: $\sup(m, D)$ 和 $|D| - \sup(m, D)$. 在之前讨论的置换过程中, 当且仅当有 s 条包含 m 的肽段被置换到 P 集合中时, P 集合中才会有 s 条肽段包含 m. 根据这个过程, 从 B_1 集合中挑选出 s 条肽段, 并把它们放入到 E_1 集合中, 此挑选方式的种数为

$$v_1(m, s) = \binom{\sup(m, D)}{s}. \tag{6}$$

然后, 从 B_2 集合中挑选出 $|P| - s$ 条肽段, 也把它们放到 E_1 集合中. 该挑选方式的种数是

$$v_2(m, s) = \binom{|D| - \sup(m, D)}{|P| - s}. \tag{7}$$

显然, E_1 集合的大小为 $|P|$. 接着, 把 B_1 和 B_2 集合中剩余的肽段都放到 E_2 集合中, 该挑选种数只有 1 种, 且 E_2 集合的大小为 $|N|$. 所以, 所有可能的 E_1 和 E_2 集合的数量为

$$v_1(m, s)v_2(m, s), \tag{8}$$

其中, E_1 和 E_2 集合中的肽段都各有 $|P|!$ 和 $|N|!$ 种排列顺序.

随后, 将 E_1 和 E_2 集合中的肽段分别放入到 P 和 N 集合中, 就得到了原始数据集合的一种置换数据集合, 且这种置换数据集合中的 P 集合中仅有 s 条肽段包含 m. 根据以上模拟过程, 能够得出 P 集合中仅有 s 条肽段包含 m 的置换数据集合的数量为

$$h_1(m, s) = v_1(m, s)v_2(m, s)|P|!|N|!, \tag{9}$$

即 m 在 P 集合中支持度为 s 的情景的频率.

假设 M 是一个包含 D 集合中所有磷酸化基序的集合, 如果让 M 中每个磷酸化基序 m 的支持度 s 从下界 $L(m)$ 递增到上界 $U(m)$, 统计度量值的总数量为

$$\sum_{m \in M} \sum_{L(m)}^{U(m)} h_1(m, s). \tag{10}$$

在磷酸化基序精确 p-value 的计算中, 式 (10) 的结果将作为分母, 一些满足条件的式 (9) 的结果的累加将作为分子. 因此 $|P|!$ 和 $|N|!$ 可以被约分掉. 为了减少计算, 将式 (9) 改写为

$$h_2(m,s) = v_1(m,s)v_2(m,s). \tag{11}$$

详细的 EPPM 算法步骤和其解释如下所示:

算法 1 EPPM (P, N, min_sup, α)

输入: 前景集合 P; 背景集合 N; 最小支持度阈值 min_sup; 置信水平 α.
输出: 统计显著的磷酸化基序集合 Result.

```
 1: Result ← ∅, T ← ∅, sum_ts ← 0;
 2: M1 ← FP-Growth (P, min_sup);
 3: M2 ← FP-Growth (D, min_sup);
 4: for each m1 ∈ M1 do
 5:     ms(m1) ← 0;
 6: end for
 7: for each m2 ∈ M2 do
 8:     for s ← L(m2) to U(m2) do
 9:         ts ← z(m2, s);
10:         fc ← h2(m2, s);
11:         T = T∪{⟨ts, fc⟩};
12:         sum_ts ← sum_ts+fc;
13:     end for
14: end for
15: sort(T);
16: accumulate(T);
17: for each m1 ∈ M1 do
18:     ms(m1) ← search(org_ts(m1), T);
19:     p(m1) ← ms(m1)/sum_ts;
20: end for
21: Result← BH(M1, α);
22: return Result.
```

(1) 在 P 集合中用 FP-growth 算法挖掘支持度超过 min_sup 的磷酸化基序, 并将其放入到 M_1 集合中, 这些基序即是需要被同时检验的磷酸化基序 (第 2 步). 然后, 在 D 集合中用 FP-growth 算法挖掘支持度超过 min_sup 的磷酸化基序, 并将其放入到 M_2 集合中, 这些基序即是所有置换数据集合中可能出现的磷酸化基序 (第 3 步).

(2) 为 M_1 集合中每个需要被检验的磷酸化基序 m_1 分配一个变量 $\mathrm{ms}(m_1)$, 该变量储存的是所有统计度量值中大于等于 m_1 原始统计度量值的统计度量值的个数 (第 4~6 步).

(3) 计算 M_2 集合中每个磷酸化基序 m_2 的每个支持度 s 所对应的统计度量值 ts 和其频率 fc, 并将每一对 $\langle \mathrm{ts}, \mathrm{fc}\rangle$ 放入到 T 集合中 (第 7~11 步). 之后, 把每一个频率 fc 累加给 sum_ts (第 12 步), sum_ts 最终的结果即是所有可能的置换数据集合中统计度量值的总数.

(4) 根据 ts 值的大小降序排列 T 集合中所有的 $\langle \mathrm{ts}, \mathrm{fc}\rangle$ 对 (第 15 步), 然后累加 T 集合中所有 $\langle \mathrm{ts}, \mathrm{fc}\rangle$ 对的 fc 值 (第 16 步). 例如, 假设 T=$\{\langle \mathrm{ts}_1, \mathrm{fc}_1\rangle, \langle \mathrm{ts}_2, \mathrm{fc}_2\rangle\}$, 且 $\mathrm{ts}_2 > \mathrm{ts}_1$, 经过 sort($T$) 之后, T=$\{\langle \mathrm{ts}_2, \mathrm{fc}_2\rangle, \langle \mathrm{ts}_1, \mathrm{fc}_1\rangle\}$; 经过 accmulate($T$) 后, T=$\{\langle \mathrm{ts}_2, \mathrm{fc}_2\rangle, \langle \mathrm{ts}_1, \mathrm{fc}_1 + \mathrm{fc}_2\rangle\}$. 排序和累加的目的是为了能够快速找到至少和某个磷酸化基序原始统计度量值一样大的统计度量值的个数.

(5) 把每个需要被检验的磷酸化基序 m_1 的原始统计度量值 org_ts(m_1) 与 T 集合中的 ts 值比较, 如果 org_ts(m_1) 值等于某个 ts$_j$ 值, 则返回该 ts$_j$ 值对应的 fc$_j$ 值给 ms(m_1) (第 17∼18 步). 接着, m_1 的精确置换检验 p-value 便可由 ms(m_1) 和 sum_ts 计算得出.

(6) 最后, 使用 BH 方法将 M_1 中的 FDR 控制在置信水平 α 下, 并将最终结果返回到 Result 集合中 (第 21 步).

3.5 加速技术

为了提升 EPPM 算法的效率, 采用以下 3 个加速技术.

(1) 分组计算. 观察发现, 给定两个磷酸化基序 m_1 和 m_2, 如果 $\sup(m_1, D)$ 和 $\sup(m_2, D)$ 相等, m_1 和 m_2 将会有相同的支持度范围, 所以磷酸化基序 m_2 的第 8∼11 步重复执行了磷酸化基序 m_1 的第 8∼11 步. 因此, 可以把 M_2 集合中支持度相等的磷酸化基序分为一组, 针对不同的组来执行第 8∼11 步, 这样便能够大量减少循环的次数. 在执行了第 3 步后, 把具有相同支持度的磷酸化基序分到一组. 每组都执行第 9 步计算其统计度量值, 但第 10 步需要改变为 fc $\leftarrow h_2(m,s)\times$size_g, 其中 size_g 指的是这个组包含的元素个数.

(2) 连续计算. 在计算 $h_2(m,s)$ 时, 需要计算许多组合式. 如果直接计算这些组合式, 将会严重增加 EPPM 算法的计算开销. 为了提升 EPPM 的性能, 在计算每个磷酸化基序 m 相邻的支持度所对应的频率时, 即在计算 $h_2(m,s)$ 和 $h_2(m,s+1)$ 时, 可以用 $h_2(m,s)$ 的结果来计算 $h_2(m,s+1)$, 具体计算过程如下:

$$h_2(m,s+1) = v_1(m,s+1)v_2(m,s+1),$$

$$v_1(m,s+1) = v_1(m,s)\frac{\sup(m,D)-s}{s+1},$$

$$v_2(m,s+1) = v_2(m,s)\frac{|P|-s}{|D|-\sup(m,D)-|P|+s+1}.$$

这项技术被称为连续计算. 显然, 连续计算能大量减少组合式的计算开销. 在计算每个磷酸化基序 m 从下界 $L(m)$ 到上界 $U(m)$ 对应的频率时, 可以首先直接计算出 $h_2(m,L(m))$, 然后使用连续计算技术计算 $h_2(m,L(m)+1), h_2(m,L(m)+2),\ldots,h_2(m,U(m))$.

(3) 界限阈值. 在计算被检验的磷酸化基序 m 的精确 p-value 时, 需要计算出所有统计度量值中至少和 m 的原始统计度量值一样大的统计度量值的个数. 换言之, 即需要找到一个 $\langle \text{ts}_j, \text{fc}_j\rangle$ 对, 其中 ts$_j$ = org_ts(m), 并返回 fc$_j$. 在排序和累加操作过后, 所有的 $\langle$ts, fc$\rangle$ 对都是按照 ts 值降序排列的, 假设磷酸化基序 m 是 M_1 集合中原始统计度量值最小的磷酸化基序, 那么一些 $\langle \text{ts}_q, \text{fc}_q\rangle$ 对, 其中 ts$_q <$ org_ts(m), 将在第 18 步中不会被查找到, 因此储存、排序和累加这样的 $\langle \text{ts}_q, \text{fc}_q\rangle$ 对是毫无意义的. 界限阈值技术使用了一个变量 min_z 来存储所有被检验的磷酸化基序中最小的原始统计度量值. 当 EPPM 算法运行了第 10 步后, 比较 ts 和 min_z 的大小, 如果 ts <min_z, 就可以跳过第 11 步, 这样就能够过滤掉无用的 $\langle \text{ts}_q, \text{fc}_q\rangle$ 对.

4 实验结果

为了比较 EPPM 和 DSP 算法 [13] 评估磷酸化基序的性能, 本文实施了一系列实验. DSP 算法是将标准置换检验直接应用到了磷酸化基序的评估中. 之所以只和 DSP 算法进行比较, 是因为 EPPM 算法的目的是为了去除标准置换检验中存在的 3 个缺点. 文献 [13] 比较了置换检验方法同其余非置

换检验方法评估磷酸化基序挖掘结果的效率, 体现了置换检验算法的优势. 若无特殊置换次数说明, DSP 算法均采用默认的 1000 次置换次数. 所有实验都是在一台配置为 3.20 GHz CPU 和 8 GB 内存的电脑上运行的.

4.1 数据集合

应用文献 [12] 中采用的 3 组数据集合来进行实验, 其中每个数据集中的每条肽段都包含 12 个氨基酸和 1 个磷酸化残基, 且该残基在该肽段的中心位置. 详细的数据集合介绍如下:

(1) 非激酶特异的磷酸化数据集 (non-kinase-specific phosphorylation data): 该组数据来源于 Swiss-Prot (release 2011_11) 数据库 [20] 和 Phospho.ELM (version 9.0) 数据库 [21]. 从 Phospho.ELM 数据库中直接提取出被标记的磷酸化肽段作为 P 集合的候选集合; 关于 N 集合的构建, 采用的是文献 [22] 提出的方法. 最后, 再从 P 和 N 集合候选集合中各抽样出 5000 条肽段来构建 P 和 N 集合.

(2) 周期蛋白依赖性激酶的磷酸化数据集 (cyclin-dependent kinases (CDK)-specific phosphorylation data): 在细胞周期运转中, 周期蛋白依赖性激酶 (cyclin-dependent kinases) 是一种非常重要的酶. 这种激酶或者随着细胞周期被激活, 或者磷酸化相应的基底使细胞周期以有序的方式进行. 与构建非激酶特性的磷酸化数据集的方法类似, 首先从 CDK 蛋白质中提取磷酸化肽段来构建前景集合 P, 然后运用文献 [23] 提出的方法来构建背景集合 N. 最终, P 集合中含有 191 条磷酸化肽段, N 集合中含有 193 条非磷酸化肽段.

(3) A 蛋白激酶的磷酸化数据集 (protein kinase A (PKA)-specific phosphorylation data): A 蛋白激酶 (protein kinase A) 是一种 cAMP-dependent 酶. 该激酶在调控蛋白质磷酸化和激活特异基因的转录中都有着非常重要的作用. 该组数据集的构建方式和 CDK 数据集的构建方式相同, 最终得到了 107 条磷酸化肽段和 110 条非磷酸化肽段.

4.2 非零性

在标准置换检验中, 如果一个需要被检验的磷酸化基序的统计度量值很大, 那么 DSP 算法计算出的该磷酸化基序的 p-value 将很可能为 0. 为了证明这个缺点, 在每个数据集不同的 min_sup 参数下运行了 DSP 算法, 实验结果如图 3 所示.

由图 3 可知, 每个数据集合在不同参数下返回的结果中都存在一定数量的 p-value 等于 0 的磷酸化基序, 其中 non-kinase-specific 数据集中 p-value 等于 0 的磷酸化基序的数量要比其余两个数据集多, 其原因可由图 4 解释. 图 4 描绘的是每个数据集中所有被检验的磷酸化基序的原始统计度量值的分布. 从图 4 中可知, CDK-specific 和 PKA-specific 数据集中被检验的磷酸化基序的原始统计度量值都比较小, 所以它们的 p-value 为 0 的概率也很小; 而 non-kinase-specific 数据集中存在一部分原始统计度量值比较大的磷酸化基序, 这些基序由 DSP 计算得出的 p-value 存在很大的概率为 0.

用 EPPM 算法在同样的参数下运行以上数据集, 得到的结果中所有磷酸化基序的 p-value 都不为 0. 原因是 EPPM 计算的是精确零分布, 即使某个被检验的磷酸化基序的原始统计度量值是最大的, 该磷酸化基序的 p-value 也不可能为 0, 因为总能找到该磷酸化基序对应的 fc_j 值作为分子来计算该磷酸化基序的 p-value.

4.3 唯一性

由于标准置换检验的随机性, 每次执行 DSP 产生的置换数据集合都不尽相同, 因此同一个磷酸

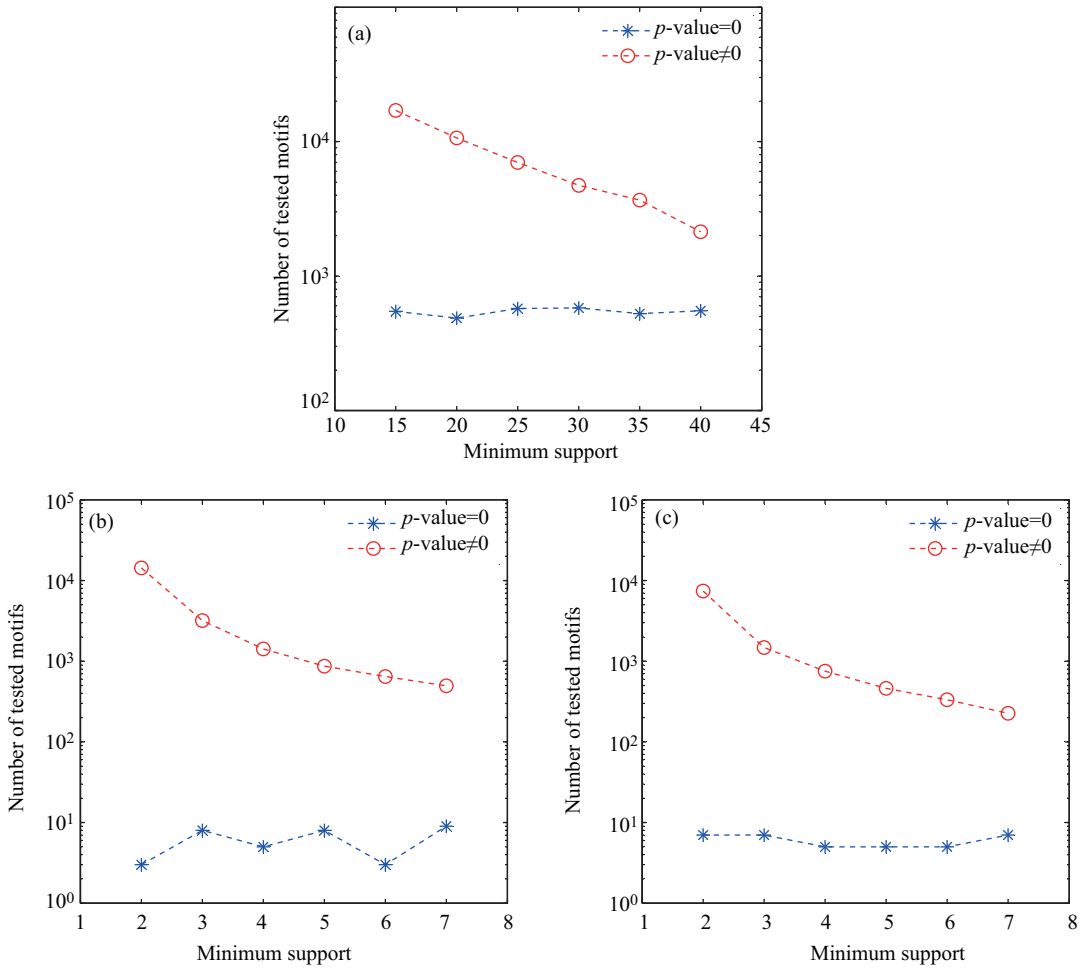


图 3 (网络版彩图) DSP 计算结果中 p-value 等于 0 和 p-value 不等于 0 的磷酸化基序的数量

Figure 3 (Color online) The number of phosphorylation motifs whose p-values are zeros and non-zeros reported by DSP. (a) Non-kinase-specific; (b) CDK-specific; (c) PKA-specific

化基序在不同次的 DSP 中计算得出的 p-value 也几乎不同. 为了证明这个缺点, 在每个数据集上执行了 100 次 DSP 和 EPPM 算法. 图 5(a) 描述了 CDK-specific 数据集中一个随机挑选的磷酸化基序在 100 次 DSP 结果中的 p-value, 从图中可以直观地看出, 每次运行结果得到的 p-value 几乎都不相同.

由于 p-value 的波动性, 导致了 DSP 每次返回的结果也不相同. 图 5(b) 给出的是在 CDK-specific 数据集上运行 100 次 DSP 和 EPPM 返回的统计显著的磷酸化基序的数量. 从图 5(b) 中也能直观地看出, DSP 返回的结果是有波动的, 但 EPPM 返回的结果是唯一的, 因为 EPPM 计算的是精确零分布, 而该分布具有唯一性, 所以在相同的数据集上多次运行 EPPM, 返回结果都是一样的.

显然, 如果 DSP 计算得出的是精确零分布的一个合理的近似零分布, DSP 返回的结果将会接近 EPPM 返回的结果. 但是因为每个数据集特性不同, 很难保证 DSP 计算得到的零分布的合理性.

4.4 精确性

DSP 如果想要得到一个更精确的结果, 需要增加置换次数. 为了证明该特性, 在 non-kinase-specific 数据集分别运行了 250, 500, 1000, 2000 和 4000 次置换的 DSP 和 EPPM 算法. 表 1 列出了该实验结

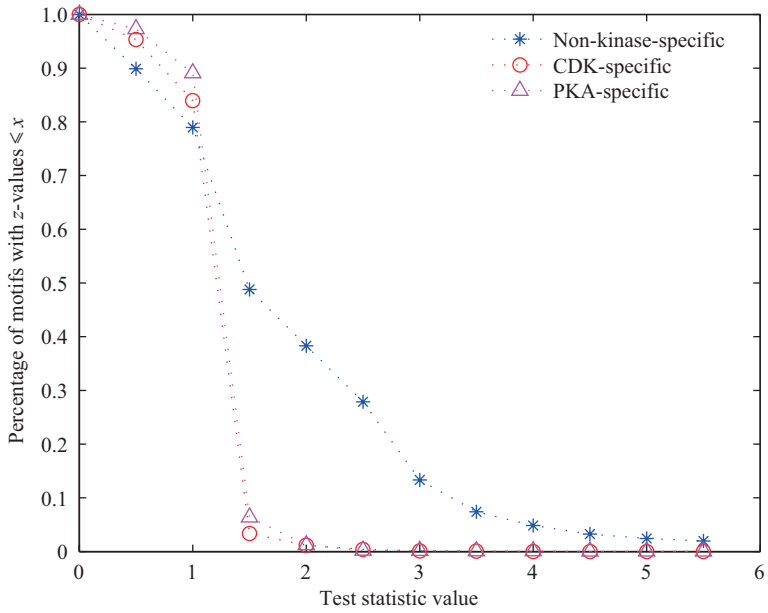


图 4　(网络版彩图) 磷酸化基序的统计度量值的分布量

Figure 4　(Color online) The distribution of the test statistic values

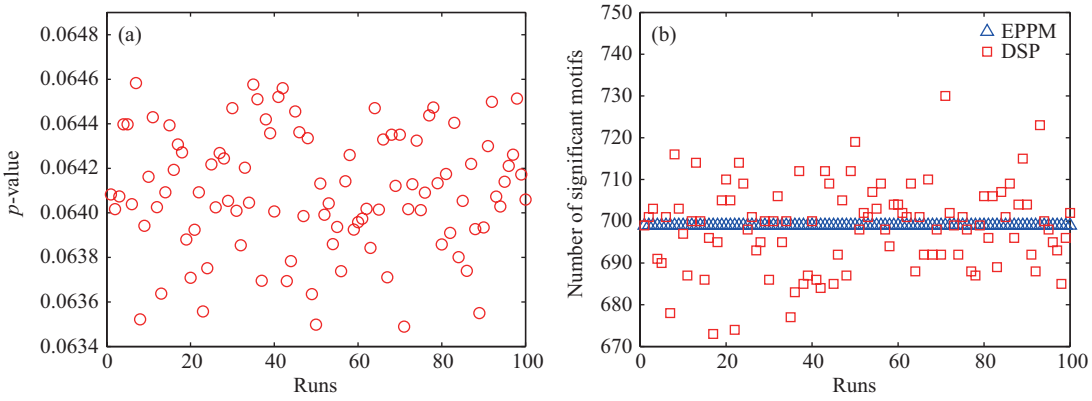


图 5　(网络版彩图) (a) CDK-specific 数据集上 100 次 DSP 运行结果中的某个随机挑选的磷酸化基序的 p-value; (b) CDK-specific 数据集上运行 100 次 DSP 和 EPPM 返回的统计显著的磷酸化基序的数量

Figure 5　(Color online) (a) The p-values of a randomly selected phosphorylation motifs with 100 different runs on CDK-specific data set reported by DSP; (b) the number of significant phosphorylation motifs returned from DSP and EPPM in 100 different runs on CDK-specific data set

果的 p-value 分布. 在表 1 中, 随着置换次数的增加, p-value 等于 0 的磷酸化基序的数量在逐渐递减, 并且 p-value 的精度在逐步递增. 例如, 置换次数增加到 2000 时, 精度可达 10^{-7}. 所以, 随着置换次数的增加, 每个磷酸化基序由 DSP 计算得出的 p-value 将会越来越接近其精确 p-value. 因为置换次数越多, 得到一个精确零分布合理的近似零分布的可能性就越大, 从而得到的结果也越稳定.

尽管增加置换次数能得到更加精确的 p-value, 但是很难确定某一个数据集需要执行多少次的置换才能得到一个可以接受的结果. EPPM 中不会存在这样的问题, 因为其计算出的就是精确的 p-value.

表 1 Non-kinase-specific 数据集上不同置换次数的 DSP 和 EPPM 结果中的 p-value 分布

Table 1 The distributions of the p-values returned from DSP with the different number of permutations and EPPM on non-kinase-specific data set

p-value	250	500	1000	2000	4000	EPPM
(0.1, 1]	3290	3290	3290	3270	3270	3270
(0.01, 0.1]	1700	1660	1660	1680	1680	1680
(0.001, 0.01]	905	900	900	889	889	889
$(10^{-4}, 0.001]$	571	566	550	559	554	554
$(10^{-5}, 10^{-4}]$	273	316	318	344	342	340
$(10^{-6}, 10^{-5}]$	135	197	216	218	225	225
$(10^{-7}, 10^{-6}]$	33	69	134	121	123	124
$(0, 10^{-7}]$	0	0	0	45	87	484
0	659	568	498	440	396	0

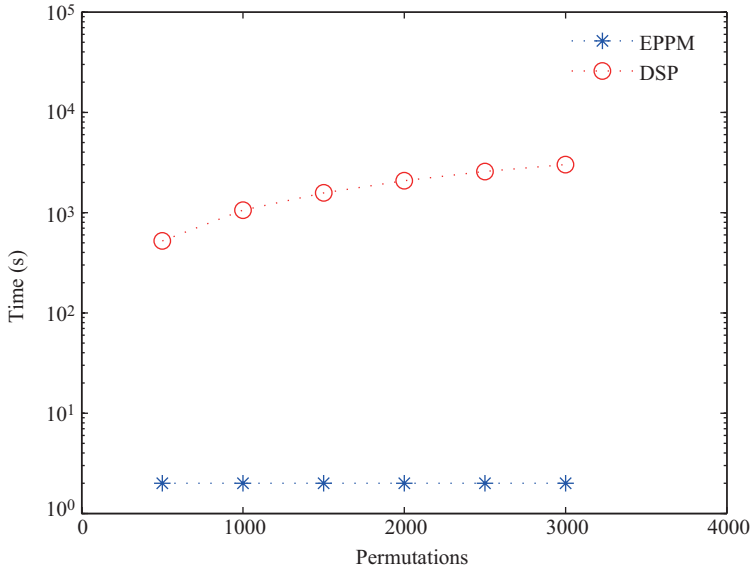


图 6 (网络版彩图) PKA-specific 数据集上不同置换次数的 DSP 和 EPPM 的运行时间

Figure 6 (Color online) The running time of DSP with different number of permutations and EPPM on PKA-specific data set

4.5 运行时间

增加置换次数可以得到更精确的结果, 但是增加置换次数也会导致更多的时间开销. 图 6 展示了 PKA-specific 数据集中不同置换次数下 DSP 和 EPPM 算法的运行时间, 显然, EPPM 的运行时间要远远低于 DSP.

图 7 给出了 DSP 和 EPPM 在各个数据集中不同 min_sup 参数下的运行时间. min_sup 越小, 挖掘到的磷酸化基序的数量就会越多. 观察图 7 发现, 需要被检验的磷酸化基序的数量对 DSP 算法的运行时间有很显著的影响. 例如, 在 CDK-specific 数据集中, min_sup=2 时的运行时间要明显高于其余 min_sup 下的运行时间. 其原因是该支持度下被检验的磷酸化基序比较多, 每次置换完成后, DSP 都

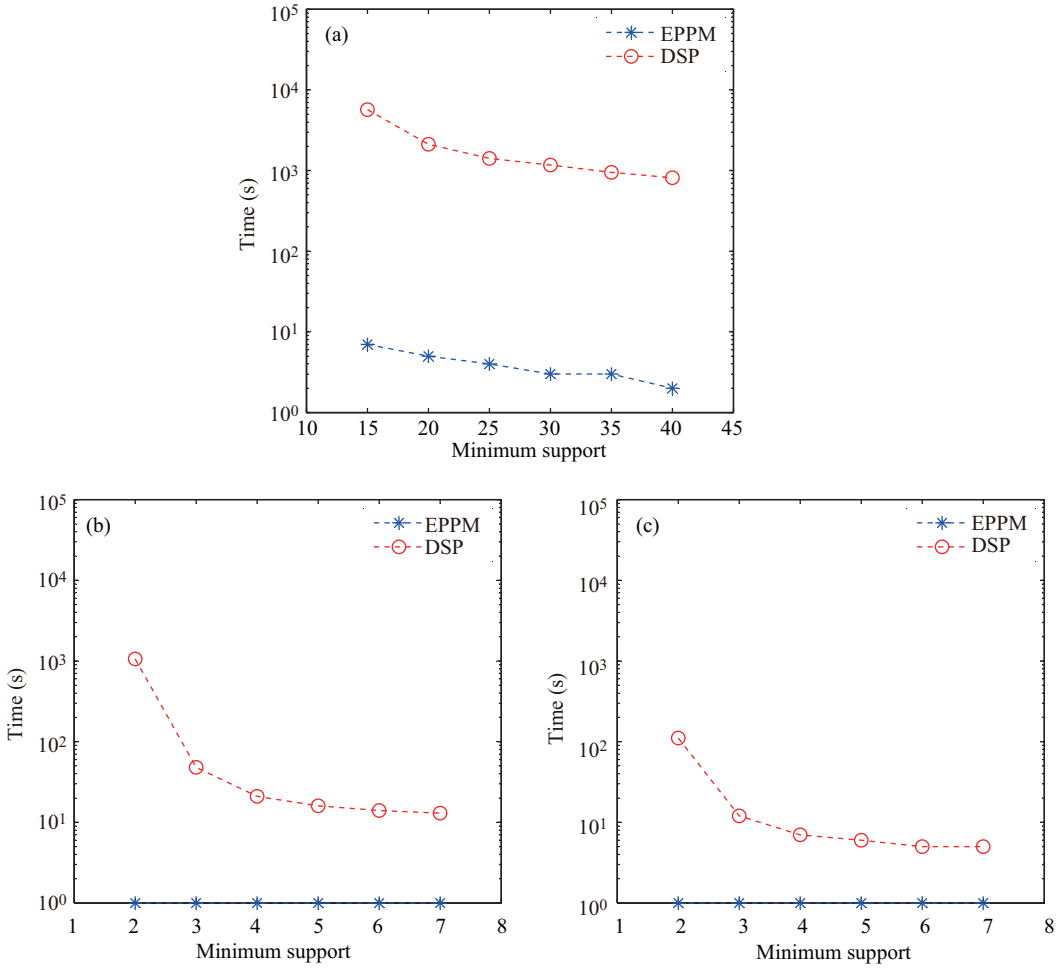


图 7 (网络版彩图) 每个数据集上 DSP 和 EPPM 在不同 min_sup 参数下的运行时间

Figure 7 (Color online) The running time of DSP and EPPM under different min_sups on each data set. (a) Non-kinase-specific; (b) CDK-specific; (c) PKA-specific

需要重新挖掘该置换数据集合中的磷酸化基序, 尽管已经采用了相对较快的 FP-growth 挖掘算法, 但挖掘过程仍然有很大的计算开销. 但是在 EPPM 中, 只需要进行一次挖掘即可, 这就大大减少了开销, 所以需要被检验的磷酸化基序的数量, 对 EPPM 算法的运行时间影响比较小.

5 结论

首先分析了精确零分布的建立可以去除标准置换检验中的 3 个缺点, 随之提出了利用 EPPM 算法来建立精确零分布, 从而被检验的磷酸化基序的精确 p-value 可由该精确零分布直接计算得到. 实验结果证明了 EPPM 算法确实能够去除掉标准置换检验中的缺点, 并且 EPPM 算法比标准置换检验算法性能更好.

观察所有 EPPM 返回的结果, 发现如果一个磷酸化基序 m 是统计显著的, 那么包含 m 的磷酸化基序也很有可能是统计显著的, 这些基序的统计显著性有很大概率是由于 m 的显著性而导致的. 因

此在计算一个磷酸化基序的精确置换检验 p-value 时, 应该去除掉其统计显著的子磷酸化基序的影响, 这方面的工作还需要进一步深入研究, 来找到解决方案.

参考文献

1 Manning G, Plowman G, Hunter T, et al. Evolution of protein kinase signaling from yeast to man. Trends Biochem Sci, 2002, 27: 514–520

2 Turk B. Understanding and exploiting substrate recognition by protein kinases. Curr Opin Chem Biol, 2008, 12: 4–10

3 Yu Y H, Yoon S, Poulogiannis G, et al. Phosphoproteomic analysis identifies Grb10 as an mTORCI substrate that negatively regulates insulin signaling. Science, 2011, 332: 1322–1326

4 Huang Z Y, Yu Y L, Fang C Y, et al. Progress in identification of protein phosphorylation by mass spectrometry. J Chinese Mass Spectrome Soc, 2003, 24: 495–500 [黄珍玉, 于雁灵, 方彩云, 等. 质谱鉴定磷酸化蛋白研究进展. 质谱学报, 2003, 24: 494–500]

5 Schwartz D, Gygi S P. An iterative statistical approach to the identification of protein phosphorylation motifs from large-scale data sets. Nat Biotechnol, 2005, 23: 1391–1398

6 Ritz A, Shakhnarovich G, Salomon A R, et al. Discovery of phosphorylation motif mixtures in phosphoproteomics data. Bioinformatics, 2009, 25: 14–21

7 Liu X Q, Wu J, Gu F Y, et al. Discriminative pattern mining and its applications in bioinformatics. Brief Bioinform, 2015, 16: 884–900

8 Chen Y C, Aguan K, Yang C W, et al. Discovery of protein phosphorylation motifs through exploratory data analysis. Plos One, 2011, 6: e20025

9 Wang T B, Kettenbach A N, Gerber S A, et al. MMFPh: a maximal motif finder for phosphoproteomics datasets. Bioinformatics, 2012, 28: 1562–1570

10 He Z Y, Yang C, Guo G Y, et al. Motif-all: discovering all phosphorylation motifs. BMC Bioinform, 2011, 12: S22

11 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases. Santiago de Chile: Morgan Kaufmann, 1994. 487–499

12 Liu X Q, Wu J, Gong H P, et al. Mining conditional phosphorylation motifs. IEEE/ACM Trans Comput Biol Bioinform, 2014, 11: 915–927

13 Gong H P, He Z Y. Permutation methods for testing the significance of phosphorylation motifs. Stat Interface, 2012, 5: 61–73

14 Han J W, Cheng H, Xin D, et al. Frequent pattern mining: current status and future directions. Data Min Knowl Discov, 2007, 15: 55–86

15 Song W, Li J H, Xu Z Y, et al. Research on a new concise representation of frequent itemset and its mining algorithm. J Comput Res Dev, 2010, 47: 277–285 [宋威, 李晋宏, 徐章艳, 等. 一种新的频繁项集精简表示方法及其挖掘算法的研究. 计算机研究与发展, 2010, 47: 277–285]

16 Han J W, Pei J, Yin Y W. Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas: ACM, 2000. 1–12

17 Neyman J, Pearson E S. On the Problem of the Most Efficient Tests of Statistical Hypotheses. New York: Springer, 1992. 23–33

18 Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc, 1995, 57: 25–133

19 Wu J, He Z Y, Gu F Y, et al. Computing exact permutation p-values for association rules. Inf Sci, 2016, 346: 146–162

20 Farriolmathis N, Garavelli J S, Boeckmann B, et al. Annotation of post-translational modifications in the Swiss-Prot knowledge base. Proteomics, 2004, 4: 1537–1550

21 Dinkel H, Chica C, Via A, et al. Phospho.ELM: a database of phosphorylation sites-update 2011. Nucleic Acids Res, 2011, 39: 261–267

22 Blom N, Gammeltoft S, Brunak S, et al. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. J Mol Biol, 1999, 294: 1351–1362

23 Dang T H, van Leemput K, Verschoren A, et al. Prediction of kinase-specific phosphorylation sites using conditional random fields. Bioinformatics, 2008, 24: 2857–2864

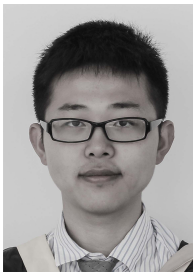
Computing exact permutation p-values for phosphorylation motifs

Jun WU[1*], Qiong DUAN[2], Lin ZHANG[1] & Zengyou HE[2,3*]

1. *School of Information Engineering, Zunyi Normal College, Zunyi* 563000*, China*;
2. *School of Software, Dalian University of Technology, Dalian* 116620*, China*;
3. *Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian* 116620*, China*
* Corresponding author. E-mail: wujun.myway@gmail.com, zyhe@dlut.edu.cn

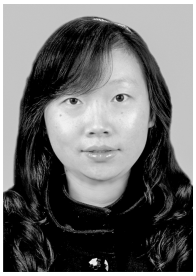
Abstract Protein phosphorylation motifs refer to position-specific amino acid patterns near phosphorylation sites. Mining phosphorylation motifs is an important task in the field of bioinformatics and several efficient methods have been proposed to uncover phosphorylation motifs. However, a large percentage of the phosphorylation motifs discovered by these algorithms are false positives. Using such motifs to perform further research will lead to inaccurate conclusions. Generally, statistical significance testing is an effective technique to filter out meaningless phosphorylation motifs. Among statistical significance testing methods, permutation testing is a commonly used method. Its usability and popularity can be attributed to its non-parametric nature. However, in permutation testing, several drawbacks narrow its range of usability. In this paper, we provide an analysis of these disadvantages and propose an algorithm called exact permutation p-values for phosphorylation motifs (EPPM) for generating an exact null distribution, from which the exact p-values of tested phosphorylation motifs can be calculated. Experiments on several datasets demonstrate that EPPM can successfully alleviate the aforementioned disadvantages and outperform the direct permutation-based method for several performance measures. To the best of our knowledge, there are still no methods in the literature that can compute exact permutation p-values for assessing phosphorylation motifs.

Keywords phosphorylation motif mining, motif assessment, statistical significance testing, permutation testing, exact permutation p-value



Jun WU was born in 1990. He received B.S. and M.S. degrees in software engineering from the Dalian University of Technology in 2013 and 2016, respectively. He is currently working at the Zunyi Normal College. His research interests include data mining and bioinformatics.

Qiong DUAN was born in 1990. He received his B.S. degree in software engineering from the Dalian University of Technology in 2015. He is currently working toward an M.S. degree at the School of Software at the Dalian University of Technology. His research interests include data mining and bioinformatics.



Lin ZHANG was born in 1984. She received her M.S. degree in software engineering from Guizhou University in 2011. She is currently working at the Zunyi Normal College. Her research interests include data mining and bioinformatics.

Zengyou HE was born in 1976. He received his B.S. degree, M.S. degree, and Ph.D. degree in computer science from the Harbin Institute of Technology in 2000, 2002, and 2006, respectively. He was a research associate in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology from February 2007 to February 2010. He is currently a professor in the School of Software at the Dalian University of Technology. His research interests include data mining and bioinformatics.