doi:10.3969/j.issn.1001-4616.2022.02.006

基于模式物种的快速同源搜索软件基准测试

王殷伟,武晶菁,张宸宁,华宜家,李　鹏,严　洁

(南京师范大学生命科学学院,江苏 南京 210023)

[摘要]　传统的 blast+软件包中的 blastp 搜索,在大数据时代下,序列搜索速度已经慢得难以接受. 同源搜索软件的开发在过去十几年取得了巨大进展,但缺乏综合的评估. 本研究对 7 个快速同源搜索软件与 blastp 进行了综合比较,结果发现,diamond 的 fast 模式总体上来说相比其他软件更快,并且有着最低的错误发现率,是追求快速搜索的最佳选择;在内存消耗上,MMseqs2 的算法在内存消耗上非常低,而 ghostx 则最高;在鉴定的 hits 数量方面,除了 blasp 以外,MMseqs2 的 s7.5 模式在中等基因组相似度 GSS 下得到的结果最多,但 s5 模式应是更好的选择. 随着 GSS 的降低,ghostx 得到的结果最多,而随着 GSS 的升高,ublast 得到的结果最多;在鉴定的 Reciprocal Best Hits(RBH)数量上,ghostx 在远缘搜索上具有优势,这一优势同样也具有共线性证据支持. 在同源搜索方面,除 ghostx 有 43.4%的额外结果外,几乎所有软件的搜索结果之间都有着很大的重叠,并且 ghostx 还有着非常低的错误发现率,而 MMseqs2 的 s3 模式却有着最高的错误发现率. 总之,MMseqs2、diamond 和 ghostx 是综合来说最好的三款替代 blastp 搜索的软件,diamond 非常适合进行直系同源推断,并且可以用“fast”模式准确地快速搜索,而“very”是权衡下来最佳的搜索模式,但如果是进行远缘物种的搜索,ghostx 则更有优势,而对于中等 GSS 下同源蛋白的鉴定,MMseqs2 的 s5 可能是更好的选择.

[关键词]　同源搜索,直系同源推断,RBH,快速算法,序列比较
[中图分类号]Q33　[文献标志码]A　[文章编号]1001-4616(2022)02-0044-08

Benchmarking Fast Homology Search Softwares Based on Model Organisms

Wang Yinwei, Wu Jingjing, Zhang Chenning, Hua Yijia, Li Peng, Yan Jie

(School of Life Sciences, Nanjing Normal University, Nanjing 210023, China)

Abstract: Blastp in the traditional blast+package has been extremely slow in the era of big data. The development of homology search software has made great progress in the past decade or so, but comprehensive evaluations are scarce. In this study, a comprehensive comparison between 7 fast homology search softwares and blastp was conducted, and it was found that fast mode in diamond is generally faster than the others and has the lowest false discover rate. In memory consumption, MMseqs2 is the lowest while ghostx is the highest. In terms of the number of identified hits, s7.5 mode in MMseqs2 had the highest number at medium Genomic Similarity Scores(GSS) except blastp, but the s5 model should be a better choice. As GSS decreases, ghostx obtains the most results, while ublast obtains the most results as GSS increases. In terms of the number of identified Reciprocal Best Hits(RBH), ghostx has an advantage in remote search, and this advantage is also supported by synteny evidence. In terms of homology search, there is a large overlap among almost all software, with the exception of ghostx, which has 43.4% additional results and the highest false discovery rate while s3 mode in MMseqs2 has the lowest. Overall, compared to blastp, MMseqs2, diamond and ghostx are the three best alternatives to blastp. Diamond is well suited for orthology inference and can search accurately and quickly in “fast” mode, and “very” is the best search mode on balance, but for remote search, ghostx is more advantageous, while for identification of homologous proteins at medium GSS, s5 mode in MMseqs2 may be a good choice.

Key words: homology search, orthology inference, RBH, fast algorithms, sequence comparision

同源搜索对比较基因组学分析十分重要,是后续诸如基因组注释、基因鉴定、基因家族聚类、GO[1]、KEGG[2]富集等一系列比较基因组学流程分析的基础. 以最流行的两个直系同源推断和基因家族聚类软

收稿日期:2021-11-18.
基金项目:国家自然科学基金项目(3167229)、江苏省高等学校自然科学研究重大项目(19KJA330001).
通讯作者:严洁,博士,副教授,研究方向:动物分子进化与系统地理. E-mail:yanjie@ njnu.edu.cn

件 OrthoMCL[3] 和 OrthoFinder[4] 为例:OrthoMCL 通常需要基于 blastp 结果进行分析,而 OrthoFinder 则是内置了 blastp、MMseqs 和 Diamond 三种搜索软件供用户选择. 直系同源被定义是一种在物种形成事件后分化的特征[5],一对在不同物种中相同的基因通常会被认定为直系同源基因. Reciprocal Best Hits(RBH)的方法因具有更少假阳性而被广泛用于直系同源基因的推断[6-8]. RBH 指两个物种中通过比对搜索软件得到最佳匹配或者最高打分的一对基因. 随着基因组时代的来临,蛋白数据呈现爆炸式的指数增长,而传统 blastp 搜索越来越难以应对快速搜索的需求,带来了对于快速同源搜索算法和软件开发的需求. 各类算法的快速同源搜索软件应运而生,但速度的提升往往会带来精度上的下降. 因此,基于时间消耗、同源对、RBH 以及软件错误率等指标对于各种同源搜索软件的评估十分重要,它决定能否找到正确的直系同源基因,决定比较基因组学分析的正确与否.

近几年来不乏有一些针对同源搜索软件的比较分析. 2014 年 Ward 等[9] 将 blast+[10] 中的 blastp 与 last[11]、ublast[12] 和 blat[13] 进行了比较,结果发现虽然 blat 速度最快,但在 RBH、同源对搜索上有着最低的预测数,并且在 RBH 推断上有着最高的错误率,而 ublast 和 last 则相比 blat 来说,有着更多的 RBH 和同源对预测数以及更低的错误率;随着衡量物种亲缘的指标基因组相似度 GSS(Genomic Similarity Scores)[14] 的降低,各个软件在 RBH 和同源对预测数上都有着随之下降的趋势,而错误率则有着随之升高的趋势. 2016 年 Saripella 等[15] 基于 16 个模式物种以及结合蛋白结构数据库信息,对基于谱(profile-based)搜索的软件 cs-blast[16]、hhsearch[17]、phmmer[18] 以及非基于谱搜索的软件 blast+中的 blastp、usearch[12]、ublast 和 fasta[19] 进行了综合评估,结果发现基于谱搜索的软件相比非基于谱的,有着更高的 AUC 值,表明其精度更高,但这同样带来了时间消耗的巨大增加. 2020 年,Hernández 等[20] 对 last、blast、diamond[21] 和 MMseqs2[22] 进行了类似于 Ward 等[9] 的研究,在不同 GSS 下得到的结果和趋势也是类似的,并且还发现 diamond 的“very”模式在速度和 RBH 结果上有着良好的平衡,diamond 是综合来说最好的软件.

尽管有了上述的一些研究,对更多优秀、快速同源搜索的软件进行比较仍然是必要的. 首先,上述的大部分软件,都会不断地进行定期更新,其运算的速率、精度是会改变的,需要进行重新评估;其次,上述的一些研究,选择的软件并不全面,一些主流和新开发的快速搜索软件并未参与评估. 纵观近十年的同源搜索软件和算法的开发和进展,选取了具有代表性的快速搜索软件,包括 usearch/ublast、last、lambda[23]、ghostx[24]、diamond、MMseqs2 以及 blast 在内,共 8 种非基于谱的软件或算法进行评估. 为何仅选择非基于谱的,这很大程度上是因为,基于谱的算法和软件,虽然带来了精度上的提升,速度却相比 blastp 有着大幅下降[15],更难实现大规模数据情况下的同源蛋白搜索. 本文旨在筛选出相比 blastp 来说更快的算法或软件,与此同时精度上有着更少下降或更高的替代品,以应对大规模数据下的同源蛋白搜索.

1 材料与方法

1.1 蛋白序列收集

研究部分参照 Saripella 等[15] 的做法,选取了 15 个具有代表性的、有一定跨度的模式物种(表 1),涵盖原核与真核生物,它们分别为,属于细菌的 *Escherichia coli* 和 *Staphylococcus aureus*,原生动物的 *Chlamydomonas reinhardtii* 和 *Dictyostelium discoideum*,真菌的 *Aspergillus nidulans* 和 *Saccharomyces cerevisiae*,植物的 *Arabidopsis thaliana* 和 *Zea mays*,无脊椎动物的 *Drosophila melanogaster* 和 *Caenorhabditis elegans* 以及脊椎动物的 *Homo sapiens*、*Danio rerio*、*Xenopus tropicalis*、*Gallus gallus* 和 *Mus musculus*,各自从 NCBI 基因组数据库中下载对应基因组序列和 gff 注释,结合基因组序列和注释信息,提取蛋白序列,而对于有着不同可变剪切转录本的基因,则保留最长的蛋白序列作为该基因的代表,因而得到对于每个物种来说都是非冗余的蛋白序列.

1.2 实验机器、软件运行以及时间与内存消耗

本研究采用软件 usearch/ublast v11.0.667_i86linux32、last 1256、lambda2 v1.9.5、ghostx v1.3.6、diamond v2.0.6.144、MMseqs2 Release 13-45111 和 blast v2.5.0,以及部分软件的不同精度,即 diamond-fast、diamond-sensitive、diamond-more、diamond-very、diamond-ultra、MMseqs-s3、MMseqs-s5 和 MMseqs-s7.5,基于一台系统为 Centos8 的中小型服务器以 4 线程以及 1e-6 的阈值设定运行(表 2),物种的蛋白集合两两比对,并且包括自身比对. 服务器 CPU 型号为 AMD Ryzen Threadripper 3970X 32-Core Processor,一共 32 核,每核两线

程,内存总大小为 120GB,使用 unix 命令“time”计算并记录每次运行的真实时间(real times),在 unix 下的“while”循环中检测监测私有内存与共有内存的总即时消耗,程序运行完成后计算平均内存消耗,每次运行时确保无其他任务运行占用计算额外资源导致时间计算出现偏差.

表 1　选择的 15 个模式物种基因组序列信息

Table 1　Genome sequence information for candidate 15 model species

物种名	分类	亚类	蛋白数	登录号
大肠杆菌 *Escherichia coli*	Prokaryotes	Bacteria	4 279	GCF_000005845.2
金黄色葡萄球菌 *Staphylococcus aureus*	Prokaryotes	Bacteria	2 767	GCF_000013425.1
莱茵衣藻 *Chlamydomonas reinhardtii*	Eukaryotes	Protists	17 742	GCF_000002595.2
盘基网柄菌 *Dictyostelium discoideum*	Eukaryotes	Protists	13 291	GCF_000004695.1
构巢曲霉 *Aspergillus nidulans*	Eukaryotes	Fungi	9 556	GCF_000149205.2
酿酒酵母 *Saccharomyces cerevisiae*	Eukaryotes	Fungi	6 016	GCF_000146045.2
拟南芥 *Arabidopsis thaliana*	Eukaryotes	Plants	27 562	GCF_000001735.4
玉米 *Zea mays*	Eukaryotes	Plants	34 337	GCF_902167145.1
黑腹果蝇 *Drosophila melanogaster*	Eukaryotes	Invertebrates	13 968	GCF_000001215.4
秀丽隐杆线虫 *Caenorhabditis elegans*	Eukaryotes	Invertebrates	20 385	GCF_000002985.6
智人 *Homo sapiens*	Eukaryotes	Vertebrates	23 088	GCF_000001405.39
斑马鱼 *Danio rerio*	Eukaryotes	Vertebrates	32 816	GCF_000002035.6
热带爪蟾 *Xenopus tropicalis*	Eukaryotes	Vertebrates	21 904	GCF_000004195.4
原鸡 *Gallus gallus*	Eukaryotes	Vertebrates	17 876	GCF_016699485.2
小家鼠 *Mus musculus*	Eukaryotes	Vertebrates	22 682	GCF_000001635.27

表 2　用于运行软件的命令行

Table 2　Command lines used to run each program

软件/模式	命令
usearch	usearch-usearch_local query.faa-db subject.faa-evalue 1e-6-blast6out outputfile-threads 4-id 0
ublast	usearch-ublast query.faa-db subject.udb-evalue 1e-6-blast6out outputfile-threads 4
blasp	blastp-query query.faa-db subject.db-evalue 1e-6-out outputfile-outfmt 6-num_threads 4
ghostx	ghostx aln-i query.faa-d subject.db-o outputfile-a 4
lambda	lambda2 searchp-q query.faa-i subject.db-e 1e-6-o outputfile-t 4
last	lastal-E 1e-6-f BlastTab+-P4 subject.db query.faa>outputfile
diamond-fast	diamond blastp-q query.faa--db subject.db-e 1e-6--out outputfile--outfmt 6--threads 4
diamond-sensitive	diamond blastp-q query.faa--db subject.db--sensitive-e 1e-6--out outputfile--outfmt 6--threads 4
diamond-more	diamond blastp-q query.faa--db subject.db--more-sensitive-e 1e-6--out outputfile--outfmt 6--threads 4
diamond-very	diamond blastp-q query.faa--db subject.db--very-sensitive-e 1e-6--out outputfile--outfmt 6--threads 4
diamond-ultra	diamond blastp-q query.faa--db subject.db--ultra-sensitive-e 1e-6--out outputfile--outfmt 6--threads 4
MMseqs2-s3	mmseqs search query.db subject.db outputfile tmp-s 3-e 1e-6--threads 4
MMseqs2-s5	mmseqs search query.db subject.db outputfile tmp-s 5-e 1e-6--threads 4
MMseqs2-s7.5	mmseqs search query.db subject.db outputfile tmp-s 7.5-e 1e-6--threads 4

1.3　同源对和 RBH 的统计计算

同源对数目即每个软件运行得到的结果数目即 hits 数,而 RBH 则是每次运行结果中每个 Query 对应最佳的匹配. 首先绘制了每个软件或不同精度对应的箱线图,然后根据 blastp 结果计算基因组相似度得分,即 GSS. 在这里,计算了每对物种包括物种本身的 GSSa[14],并且以相对于 blastp 的时间、同源对数目、RBH 分别与 GSS 作误差棒图. 还计算了每种具有共线性顺序支持的可信 RBH 数目,具体来说,如果一个物种相邻的两个基因 a 和 b,与另一个物种相邻的基因 a′、b′相比,a 和 a′为 RBH,b 和 b′为 RBH,那么这两个 RBH 都为真阳性结果[7,25-26],在这里称之为可信 RBH,在本文暂称之为 CRBH(Credible RBH),同样,也绘制了 CRBH 与 GSS 的误差棒图,最后绘制了 6 种有较好表现软件结果的韦恩图,所有绘图、统计计算均在 Python 3.6 下进行,以 Matplotlib 模块绘制图片.

1.4　计算错误发现率

为了评估不同模型或程序得到结果的准确性,首先用 InterproScan v5.8[27] 软件的 superfamily v1.75[28] 蛋白家族结构数据库对所有蛋白进行了注释,接下来对所有程序运行结果在不同期望阈值下的错误发现率进行统计计算. 具体来说,如果程序搜索匹配得到的一对蛋白,有着完全相同的 superfamily 注释结果,

则为阳性匹配;有着部分相同的 superfamily 注释结果,则为模糊匹配,不参与后续计算;有着完全不同的 superfamily 注释,则为阴性匹配,错误发现率则等于在不同期望阈值下的假阳性结果总数除以所有阳性结果的总数.

2 结果与讨论

2.1 运行时间

将所有程序运行计算得到的时间与 blastp 相除,来观测其各自相对 blastp 所节省的时间. 结果发现,大部分程序运行时间平均数不到 blastp 时间消耗的 5%,而 last、usearch 和 diamond-fast 运行时间平均数则不到 blastp 时间消耗的 2.5%,是速度最快的 3 个程序(图 1). 虽然从图中看似并无太大差异,但配对样本 t 检验表明,3 个程序在时间消耗节省上来说,diamond-fast<last<usearch(p-value<1e−5). 所有程序的运行时间均小于 blastp,除了 MMseqs2 的 s7.5 精度模式,有少部分离群点(未展示),速度慢于 blastp.

除了关注不同软件间的比较,还关注了相同软件不同精度下的运行效率,即 diamond 和 MMseqs2 的不同精度模式下的表现.

对于 diamond 来说,研究发现,与之前的研究类似,“sensitive”、“more”和“very”的运行时间几乎无太大差异[20],因此,在选择这三种选项时,理论上来说一定是精度更高的“very”模式更好,而“fast”相比这三种精度下的运行速度明显更快,“ultra”相比则更慢.

对于 MMseqs2 来说,不同的精度下,速度有着明显的差异,尤其是 s7.5 精度模式,时间消耗已经高于除了 blastp 外的所有程序,并且在一些运行中,速度慢于 blastp,这体现了其运行时间的巨大变异性和不稳定性,如果该精度下后续的评估中并没有发现随之带来的各种指标评估下的良好改进,那么该精度模式则是个不被推荐的选项.

本研究并未发现不同程序时间消耗与 GSS 之间的关联性(图 2),但可以发现的是,MMseqs 的 s7.5 精度模式的相对耗时在不同 GSS 下波动较大,呈现“中间低,两头高”的模式,diamond 的“ultra”模式也呈现类似模式,但较前者更加平稳些,其他软件之间的波动以及趋势并不完全一致,但总体呈现在不同 GSS 下的平稳均匀分布,这说明大部分软件每次实验相对于 blastp 节省时间的比例是一个稳定的小区间,换言之,最不稳定的 MMseqs 的 s7.5 精度模式,如果在精度上没有表现出巨大优越性,那么其在大规模同源搜索时就不是一个良好的选择.

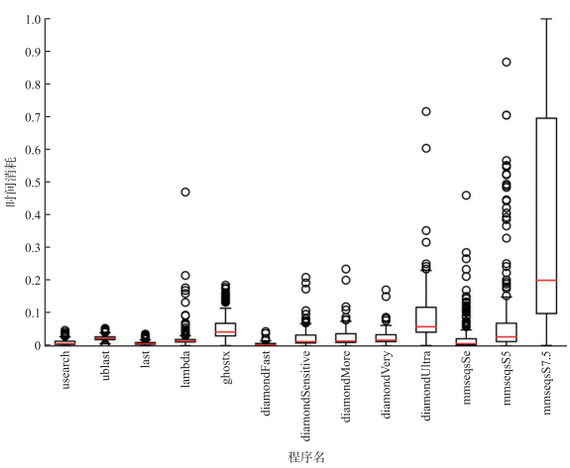

图 1 不同程序每次同源搜索相对 blastp 运行速度的差异

Fig. 1 Differences in the speed of homologous protein search relative to blastp by different programs

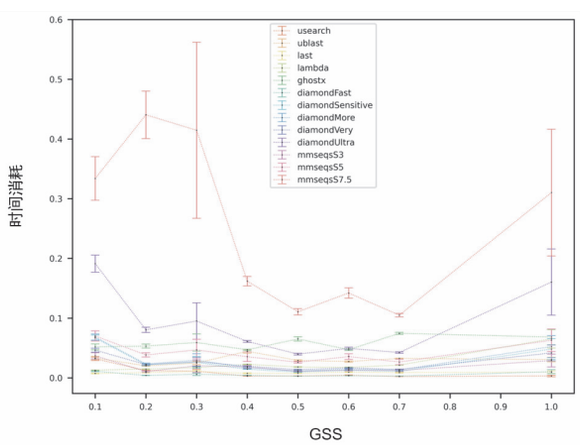

图 2 不同程序在不同 GSS 下的相对于 blastp 的时间消耗的误差棒图

Fig. 2 Error bar graph of time consumption relative to blastp for different programs at different GSS

2.2 运行内存

同样如上,将所有程序计算得到的平均内存消耗与 blastp 相除,得到相对 blastp 的内存消耗. 结果显示(图 3),除了 MMseqs2,其余所有程序的相对运行内存都显著高于 blastp,其中 ghostx 最为显著,其平均运行内存消耗约为 blastp 的 22 倍,这表明在运行 ghostx 进行同源搜索时,尤为需要注意可用内存空间的

大小. 其次比较高的是 diamond 的“ultra”精度模式,而 usearch、ublast、last 和 lambda 则有着比 blastp 较高但接近的内存占用. 值得注意的是,这些结果在大体上与时间消耗表现一致,这可能表明有些内存占用较少的软件可能不需要太大的开销进行更多的搜索,从而在时间消耗上表现较低.

当关注到相同软件不同精度下的内存占用时,MMseqs2 三种精度模式下相对平均内存消耗的分布并无太大差异,但明显低于其他程序,包括 blastp(配对样本 t 检验,p-value<1e−5),而 diamond 则随着精度的提高,呈现出明显的内存消耗提升.

2.3　同源蛋白对

与时间消耗误差棒图(图 2)一致的是,研究发现,不同软件在不同 GSS 下鉴定出的相对于 blastp 的同源蛋白数的数目分布也呈现“中间低,两头高”的趋势(图 4),并且总体分布趋势与时间消耗图吻合,这表明程序能够鉴定出的结果数目的大小和时间是有一定关联性的,这也与预期和直觉一致,结果数目越多,时间消耗越大.

几乎所有软件在不同 GSS 下鉴定出的同源蛋白对都要少于 blastp 的结果. 唯一的例外是 ghostx 在低 GSS 的情况下获得的结果大大增加,最多可达 blastp 结果的 3 倍以上. 而 ublast 则在很高 GSS 的情况下,鉴定的结果数目越为接近 blastp,但大部分情况下,MMseqs 的 s7.5 和 s5 模式有更多的结果数. 可以看到 MMseqs 的 s7.5 精度模式在时间消耗上的提升确实带来了鉴定同源蛋白对数目上的提升,但提升并不明显:在低 GSS 下显著低于 ghostx,在高 GSS 下的则略少于 ublast. 如果仅从同源蛋白结果数目上来看,ghostx 适用远缘搜索,ublast 适用于近缘搜索. 在中等 GSS 的情况下,MMseqs 的 s5 精度模式获得结果的数目仅次于 s7.5 精度模式,但考虑到时间消耗,并且 s5 与 s7.5 的数目差异并不太大,MMseqs 的 s5 精度模式是更好的选择.

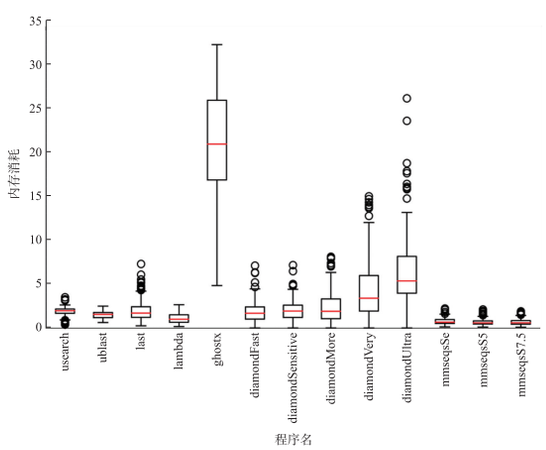

图 3　不同程序每次同源搜索相对 blastp 运行内存消耗的差异

Fig. 3　Differences in memory consumption of homologous protein search relative to blastp by different programs

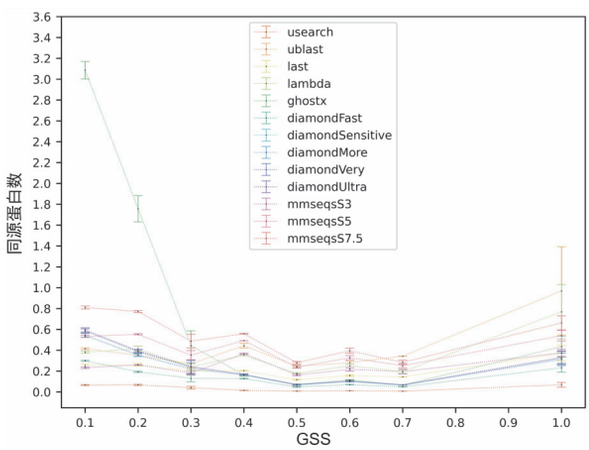

图 4　不同程序在不同 GSS 下鉴定出的相对于 blastp 的同源蛋白数的误差棒图

Fig. 4　Error bars of the number of homologous proteins identified by different programs relative to blastp at different GSS

2.4　RBH

除了考虑同源蛋白数目上的评估,对于 RBH 的评估也是非常重要的,因为前者关联着基因鉴定,而后者则关联着直系同源推断,两者都是比较基因组分析的重要步骤. 不同于相对时间消耗以及同源蛋白与 GSS 不明显的关系模式,不同程序鉴定出的 RBH 相对于 blastp 鉴定出的数量,在大部分软件中都呈现出了随着 GSS 降低而降低的趋势(图 5),而 MMseqs 的 s7.5 精度模式,则在所有 GSS 下与 blastp 数目保持一致和稳定. 令人惊讶的是,diamond 的“ultra”、“sensitive”、“more”和“very”的 RBH 数目,尽管在低 GSS 下有所波动下降,但整体都稳定在 blastp 结果数目的约 90%左右,而 ghostx 则呈现出随着 GSS 降低相对 RBH 数显著增高的趋势,这同样也表现出了 ghostx 在远缘搜索的相对优势. 结合相对时间消耗来考虑,这些结果表明,在进行远缘搜索进行直系同源推断的时候,在不考虑错误率的情况下,ghostx 是一个良好的选择,可以获得最多的结果数目,而在其他情况下,考虑到 MMseqs7.5 的耗时之多,以及 diamond 除了“fast”外其他精度结果数目的接近以及“sensitive”、“more”和“very”时间消耗的接近,diamond 的“very”仍

然是一个综合来说非常优秀的运行模式和优先考虑的选择.

然而,单纯从数量上来评估 RBH 鉴定及直系同源推断能力的优秀与否是不可行的,因为更多的 RBH 有可能引入更多的假阳性结果,因此需要对 RBH 评估的错误率进行推断,之前的两个研究都是考虑共线性关系,以及旁系同源关系,来计算错误率[9,20].

然而,不同软件鉴定的旁系同源基因,仍然有假阳性的可能,这里考虑以 CRBH 进行比较,即仅比较相对 blastp 来说,具有相邻共线性位置关系支持的可信 RBH 数目,因而避免其他噪声. 结果发现,在高 GSS 的情况下,各程序 CRBH 数目与 blastp 结果数目差别不大,而随着 GSS 的降低,各程序的相对 CRBH 出现了明显的分歧(图 6). ghostx 同样在 CRBH 上,体现着其在远缘搜索的优势,这表明,ghostx 得到的 RBH、同源蛋白数目在远缘中的增加,同时确实会带来真实的、可靠的 RBH 结果的增加,而其次优秀的就是 diamond 的“ultra”、“sensitive”、“more”和“very”模式,在 GSS 降低的情况下,也展现出了比 blastp 更多的结果,并且彼此之间的差异不大,这说明,从 CRBH 上来考虑,diamond 的“very”精度模式仍然是一个良好的选择.

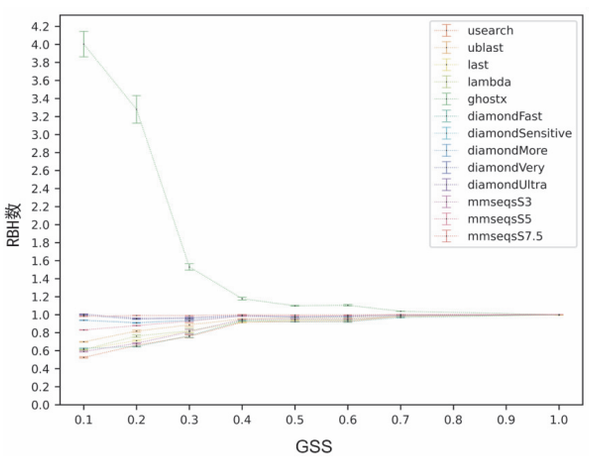

图 5　不同程序在不同 GSS 下鉴定出的相对于 blastp 的 RBH 数的误差棒图

Fig. 5　Error bars of the number of RBH identified by different programs relative to blastp at different GSS

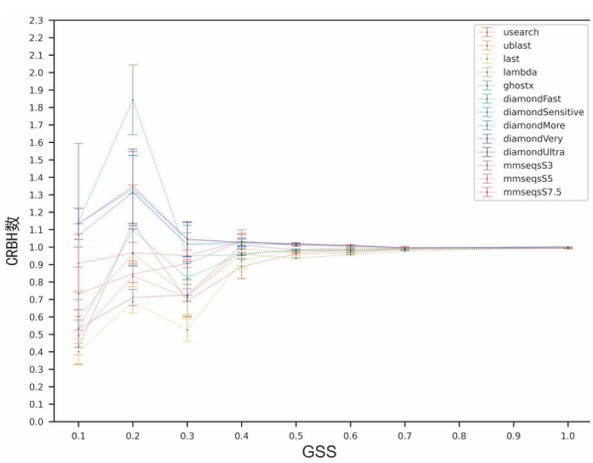

图 6　不同程序在不同 GSS 下鉴定出的相对于 blastp 的 CRBH 数的误差棒图

Fig. 6　Error bars of the number of CRBH identified by different programs relative to blastp at different GSS

2.5　代表性软件交集评估

对于各个软件的结果重合情况评估,也是非常重要的,越多的重合也同样表明了结果的可靠性. 为了使得绘图具有可读性,并且考虑到 usearch 和 last 在除了时间消耗之外其他指标上的评估表现较差以及 MMseqs2 和 diamond 的更高精度理论上应有更好的结果表现,仅选取了 blastp、diamond 的“ultra”精度模式、MMseqs2 的 s7.5 精度模式、ublast 和 lambda 作为代表,进行评估.

首先统计、绘制同源蛋白对结果的韦恩图(图 7),如图所示,大部分软件之间所鉴定出的同源蛋白对之间都有着或多或少的重合,所有软件的重合仅有 4.8%,大部分软件都与 blastp 之间有着重合,这些重合结果具有可靠性. ghostx,则有着 22.0%的最多特有搜索结果,这表明其与其他软件搜索结果交集比较少,原因正如前文所述,其鉴定出了更多的同源蛋白对、更多的 RBH、更多的 CRBH;而 diamond 的“ultra”精度模式仅有 0.1%的特有搜索结果,这表明其大量的搜索结果与其余软件都有交集,表明了其结果的可靠性,diamond 软件的优秀之处又在此处展现. ublast 和 lambda 这些更加快速但精度相较来说更低、搜索结果更少的软件,特有的结果分别仅有 5.6%和 3.0%,这可能表明,这些快速搜索软件能够快速搜索出一些与其他软件重合的、可靠的同源搜索结果,然后快速结束搜索,保留下少部分可靠的结果,至少在其鉴定出的结果上来说,是比较可靠的.

接下来绘制了不同代表性软件的 RBH 结果的韦恩图(图 8),可以明显看出,相比同源蛋白鉴定结果在各个软件中的差异性,RBH 结果的差异性明显更少,所有软件的共有的 RBH 占到了所有结果的 21.7%,这同样也是可以预期的,因为真实的直系同源对相较于其他的同源基因对,往往会有着最高的相似性,从而在软件搜索中获得最高得分而被保留. diamond 的“ultra”精度模式和 MMseqs2 的 s7.5 模式所拥

有的特有 RBH 最少,约为 1.2%,但是仍然需要速度上的考量,diamond 在快速预测出可靠的、准确的 RBH 上具有优势. 同样符合预期的是,在远缘预测以及 RBH 数量、CRBH 数量上具有明显优势的 ghostx,具有显著最多的特有 RBH 预测占比,达到了约 43.4%,lambda 和 ublast 在特有的搜索结果上占比同样较低,分别有 5.4%和 3.8%.

总而言之,6 种代表性软件统计、绘制的韦恩图与预期相符,ghostx 在远缘搜索上能够鉴定出更多的同源蛋白对、RBH 和 CRBH,因此会有着更多的特有鉴定结果,而 diamond 软件鉴定结果可靠性,也体现在其与各个软件结果都互有交集,特有结果数目较少上.

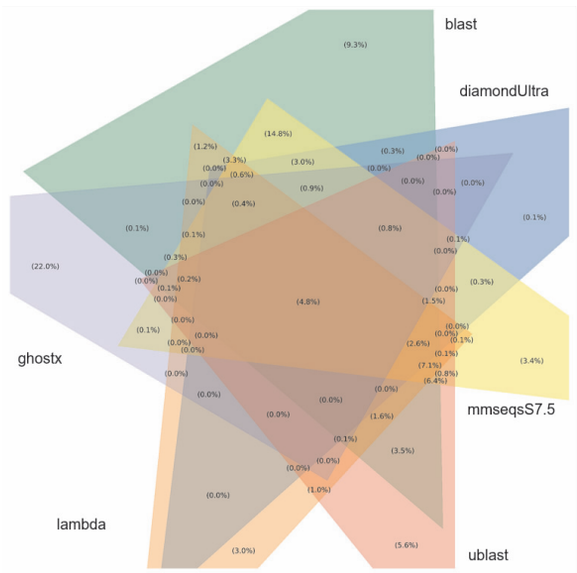

图 7　代表性软件鉴定的同源蛋白韦恩图

Fig. 7　Venn diagram of homologous proteins identified by representative software

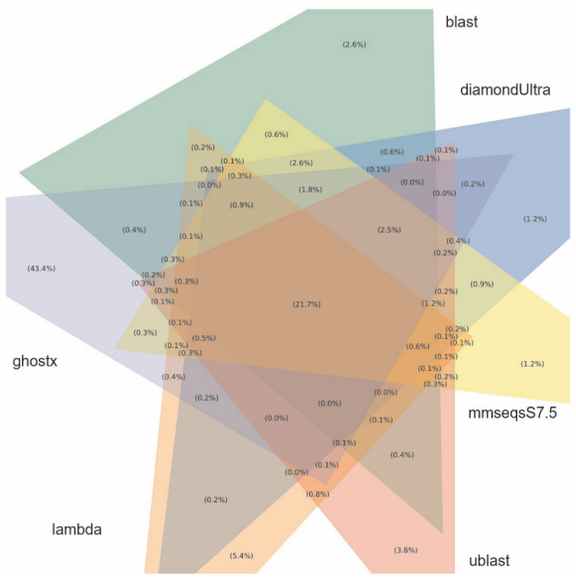

图 8　代表性软件鉴定的 RBH 韦恩图

Fig. 8　Venn diagram of RBH identified by representative software

2.6　错误发现率

为了对不同软件程序的准确性进行更进一步的评估,根据 superfamily 数据库对所有搜索蛋白的注释结果,计算了不同期望阈值下的错误发现率(图 9). 结果发现,ghostx、MMseqs2 的 s5、s7.5 以及 diamond 所有精度下的错误发现率均低于 0.01,blastp 的则低于 0.02,lambda、ublast 和 usearch 在低期望阈值(小于 1e−10)下能够将错误发现率总体控制在 0.05 以下,而 last 和 MMseqs2 的 s3 的错误发现率比较高,last 总

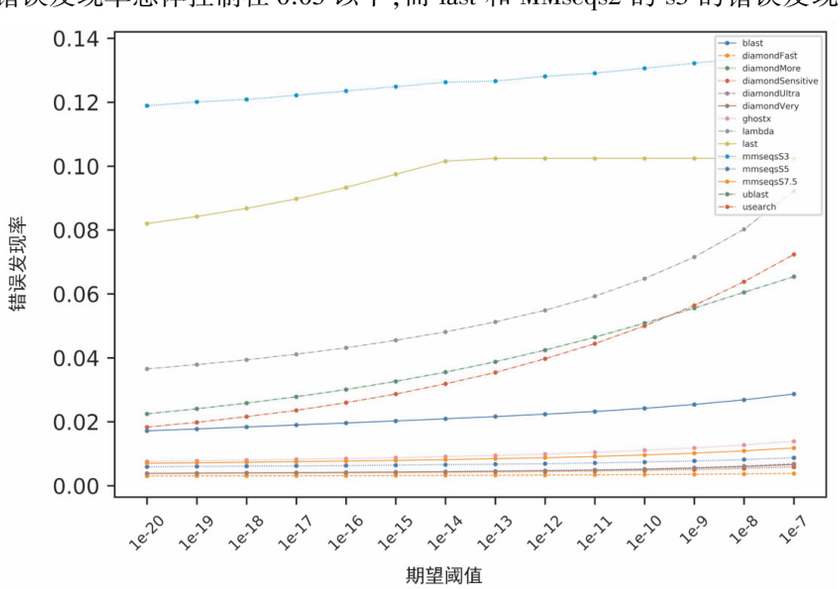

图 9　不同程序搜索结果的错误发现率

Fig. 9　False discovery rate of search results for different programs

体在 0.1 以下但高于 0.08,MMseqs2 的 s3 则表现最差,总体在 0.14 以下但高于 0.12,这可能表明 s3 模式相对来说要尽量避免使用.

值得注意的是,虽然从直觉上来说时间消耗较少、精度模式选择较低的软件应该有着很高的错误率,但实际上并非如此,diamond 和 MMseqs2 的不同精度模式就是很好的例子. diamond 的"fast"精度模式下的错误发现率是最低的,这表明其虽然进行快速搜索后,在数量上要少于高精度的搜索,但其至少能保证得到的结果是准确的,这与 MMseqs 的 s3 模式是相对的. blastp、ghostx、MMseqs2 和 diamond 高精度的搜索,在获取的结果数量上高于其他快速搜索软件的同时,也能将错误发现率控制在较低的值,可见这些软件的可靠性和优秀性.

3 结论

本研究选取了共 7 种快速同源搜索软件或程序 usearch/ublast、last、lambda、ghostx、diamond、MMseqs2,包括 diamond 的 5 种不同精度模式"fast"、"sensitive"、"more"、"very"、"ultra"以及 MMseqs2 的 3 种不同精度模式 s3、s5、s7.5,与 blastp 在时间消耗、同源蛋白对、RBH、CRBH、重合状况以及错误发现率上进行综合比较,来选择在不同状况下对大数据进行搜索的 blastp 的替代品. 结果表明,如果追求速度同时保证准确性,diamond 的"very"精度模式是最佳选择,因为其有着最低的错误发现率以及最快的搜索速度,而 MMseqs 的 s3 精度模式则有着最高的错误发现率,可能需要避免使用;在进行远缘物种同源搜索、直系同源推断时,ghostx 由于能够得到更多的直系同源对、RBH、CRBH 以及适中的时间消耗节省和非常低的错误发现率,成为最佳的选择,尽管其有着最高的内存消耗;而只有在进行近缘物种搜索时,快速搜索软件 ublast 能得到更多的同源蛋白对结果,在大部分的 GSS 下,对于同源蛋白的搜索与鉴定,MMseqs7.5 与 MMseqs5 差异不大,且两者错误发现率均低于 0.01,都是良好的两个选择,但考虑到时间消耗,MMseqs5 应是更好的选择;如果是应对于进行直系同源推断的研究目的需求,diamond 是综合来说最佳的软件,并且其最适合以"very"精度选项运行,能够得到速度与精度良好的权衡. 本研究为不同目的下选择和使用不同快速搜索软件提供了参考和指南.

[参考文献]

[1] CONSORTIUM G O. The Gene Ontology(GO)database and informatics resource[J]. Nucleic acids research,2004,32(suppl_1):D258-D261.

[2] KANEHISA M,GOTO S. KEGG:kyoto encyclopedia of genes and genomes[J]. Nucleic acids research,2000,28(1):27-30.

[3] LI L,STOECKERT C J J R,ROOS D S. OrthoMCL:identification of ortholog groups for eukaryotic genomes[J]. Genome research,2003,13(9):2178-2189.

[4] EMMS D M,KELLY S. OrthoFinder:solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy[J]. Genome biology,2015,16(1):157.

[5] FITCH W M. Homology a personal view on some of the problems[J]. Trends in genetics,2000,16(5):227-231.

[6] KRISTENSEN D M,WOLF Y I,MUSHEGIAN A R,et al. Computational methods for Gene Orthology inference[J]. Briefings in bioinformatics,2011,12(5):379-91.

[7] MORENO-HAGELSIEB G,LATIMER K. Choosing BLAST options for better detection of orthologs as reciprocal best hits[J]. Bioinformatics,2008,24(3):319-324.

[8] WOLF Y I,KOONIN E V. A tight link between orthologs and bidirectional best hits in bacterial and archaeal genomes[J]. Genome biology and evolution,2012,4(12):1286-1294.

[9] WARD N,MORENO-HAGELSIEB G. Quickly finding orthologs as reciprocal best hits with BLAT,LAST,and UBLAST:how much do we miss? [J]. PLoS one,2014,9(7):e101850.

[10] CAMACHO C,COULOURIS G,AVAGYAN V,et al. BLAST+:architecture and applications[J]. BMC bioinformatics,2009,10(1):421.

[11] KIEŁBASA S M,WAN R,SATO K,et al. Adaptive seeds tame genomic sequence comparison[J]. Genome research,2011,21(3):487-493.

(下转第 80 页)