基于同源性及亲缘性的必需基因预测软件Geptop改进实现

罗森

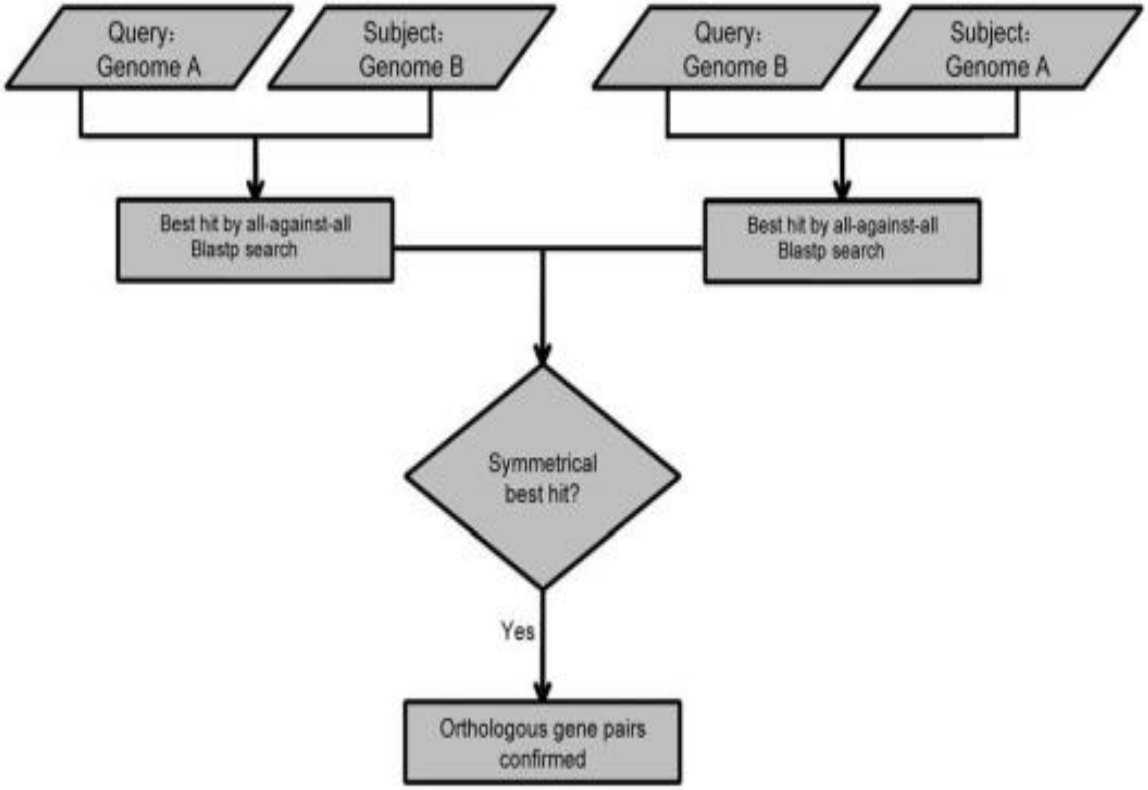
Geptop

- Geptop(gene essentiality prediction tool based on orthology and phylogeny)是由我们课题组的魏闻师兄基于基因同源性及亲缘性开发的必需基因预测工具。虽然只用到了同源性与亲缘性这两种特性，但是预测的准确性非常高，达到国际领先水平。并且在2013年成功于PLos One上发表了文章(Geptop: a gene essentiality prediction tool for sequenced bacterial genomes based on orthology and phylogeny)。

Geptop原理

- 同源性

直系同源基因的确
best hit；RBH）方

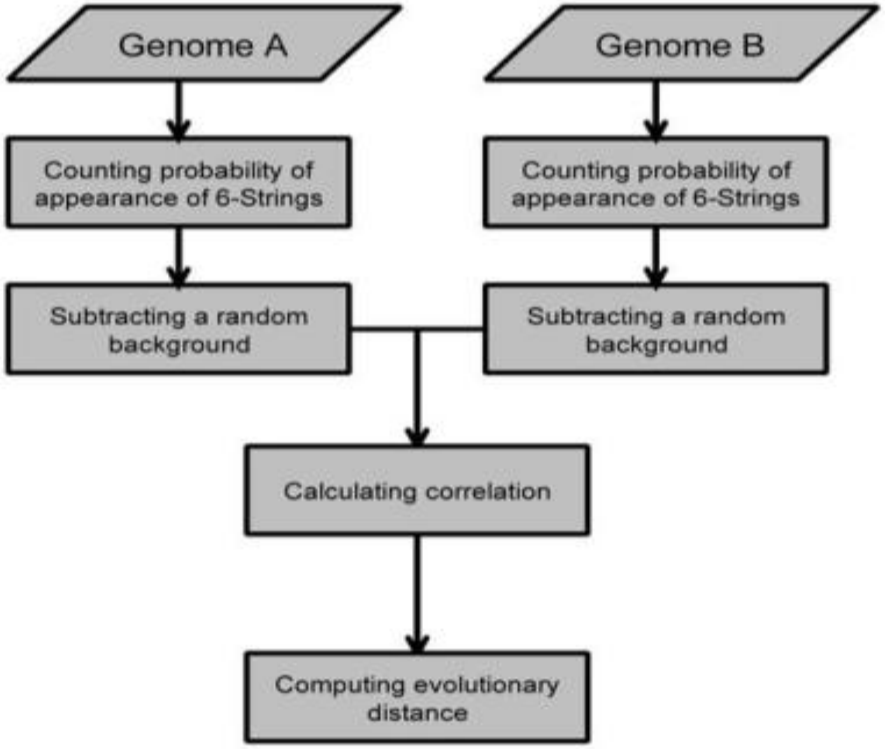


对两个特定的基因组，互相用做参考集，然后用blastp在默认参数下寻找同源基因对。如果有多重匹配，使用最小E值的为最优匹配。最终得到两组同源基因对，我们保留两组的交集为双向最优匹配的同源基因对。

Geptop原理

- 亲缘性

对于两个不同物种之间的亲
郝柏林院士提出的组成向量(
方法来实现的。计算的流程



对两个特定的基因组，我们首先算出氨基酸序列（长度为L）中六肽（Six-peptides）的出现频率，因为常用氨基酸为20种，我们得到两个6^{20}维度的频率向量$f(a_1a_2a_3a_4a_5a_6)$。

Geptop原理

当我们得到频率向量后，将此向量除以L-5得到概率向量$p(a_1a_2a_3a_4a_5a_6)$。用同样的方法计算四肽和五肽的概率向量，并使用马可夫模型（Markov model）定义随机背景$p^0(a_1a_2a_3a_4a_5a_6)$,公式如下：

$$p^0(a_1a_2a_3a_4a_5a_6)=\frac{p(a_1a_2a_3a_4a_5)p(a_2a_3a_4a_5a_6)}{p(a_2a_3a_4a_5)}$$

之后通过以下公式(公式 2-2)获得组成向量 a:

$$a(a_1a_2a_3a_4a_5a_6)=\begin{cases}\dfrac{p(a_1a_2a_3a_4a_5)p(a_2a_3a_4a_5a_6)}{p(a_2a_3a_4a_5)} & p^0\neq 0\\ 0 & p^0=0\end{cases}$$

Geptop原理

通过之前的公式计算出两个基因组的组成向量a和b，然后计算出a和b之间的余弦值C，公式如下所示：

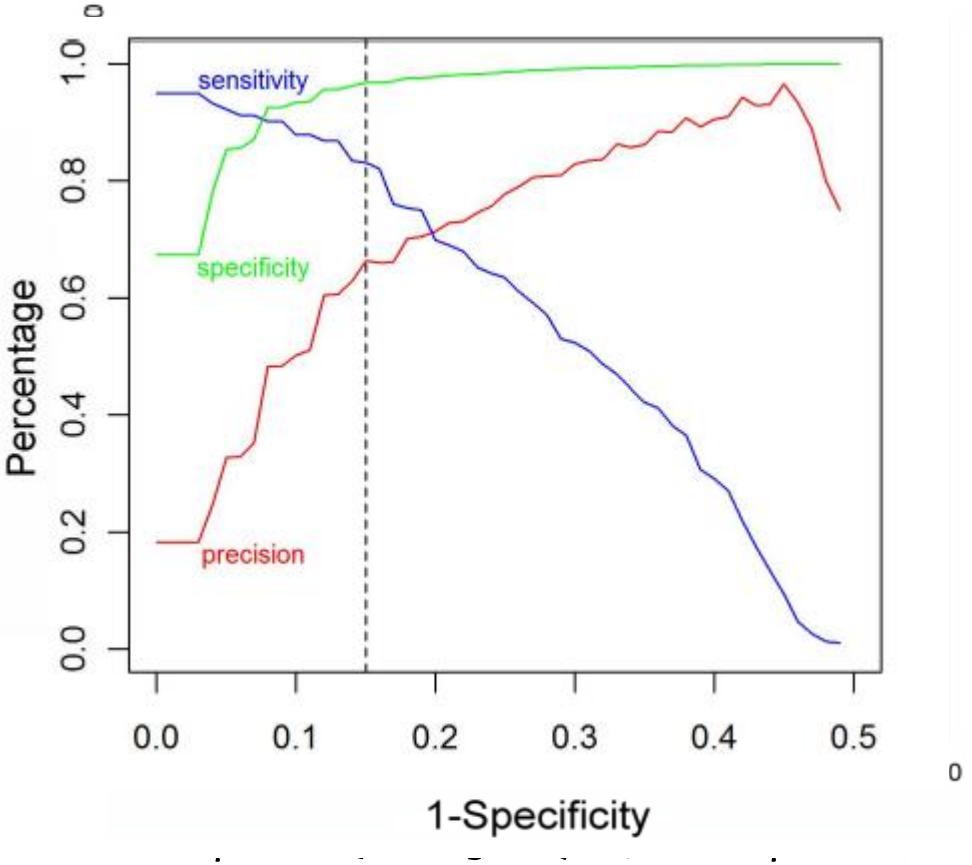
$$C = \frac{a \times b}{\|a\| \times \|b\|}$$

计算出余弦值C后，就能通过如下公式计算出两个物种之间的亲缘距离D：

$$D = \frac{1 - C}{2}$$

Geptop数据来源

- Geptop的参考集包含了19个物种，这19个物种的序列数据来自于NCBI的Genbank (http://www.ncbi.nlm.nih.gov/genbank/)，而相关的必需基因数据则来自于天津大学的DEG数据库 (http://tubic.tju.edu.cn/deg/)。



：

$_i \times D_i)$

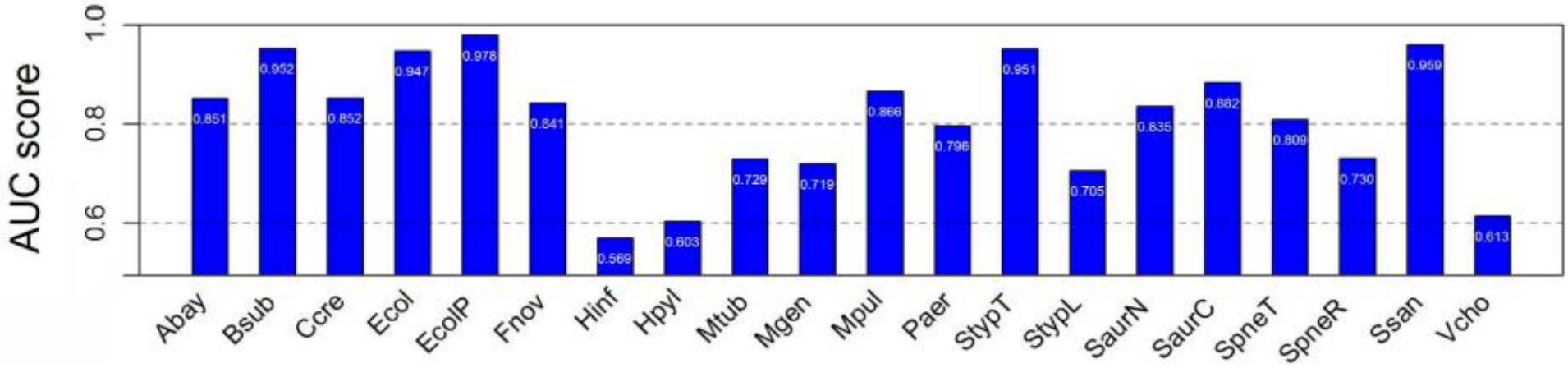
,0或1，当待测物种基因与且该基因必需基因时，M集物种数量，D为待测物距离。

- 如图所示，我们以E.coli作为测试物种，剩余18个物种作为参考集做的跨物种交叉检验的ROC曲线图。可以看出，当S值在0.15到0.2之间时拥有较高的准确率(E.coli的AUC达到了0.947)

$$F-measure=\frac{2\times sensitivity\times precision}{sensitivity+precision}$$

Geptop的评测

- 我们通过参考集的物种之间进行跨物种交叉检验来对Geptop进行评测。预测的AUC结果如下图所示：



从图中我们可以看出，大部分物种的AUC均是不小于0.8的，甚至有几个是超过了0.9，而剩下的物种中，低于0.7的也只有三个，说明Geptop预测必需基因的效果非常好。

Geptop的服务

- Geptop提供了网上在线以及本地版两种服务

- 在线网址

  http://cefg.uestc.edu.cn/geptop/

  进入网页后，上传待测物种的faa文件后，服务器会进行预测并在结束后将结果发送到你的邮箱中

- 本地服务

  你也可以在geptop网页上下载geptop的本地运行程序，按照操作说明进行本地必需基因预测

Geptop的改进

dataset1	Acinetobacter_ADP1
dataset2	Bacillus_subtilis_168
dataset3	Bacteroides_thetaiotaomicron_VPI
dataset4	Burkholderia_thailandensis_E264
dataset5	Caulobacter_crescentus_NA1000
dataset6	Escherichia_coli_K_12_substr__MG1655
dataset7	Francisella_novicida_U112
dataset8	Mycoplasma_pulmonis_UAB_CTIP
dataset9	Porphyromonas_gingivalis_ATCC_33277
dataset10	Pseudomonas_aeruginosa_UCBPP_PA14
dataset11	Salmonella_enterica_serovar_Typhimurium_SL1344
dataset12	Salmonella_enterica_serovar_Typhi_Ty2
dataset13	Shewanella_oneidensis_MR_1
dataset14	Sphingomonas_wittichii_RW1
dataset15	Staphylococcus_aureus_N315
dataset16	Staphylococcus_aureus_NCTC_8325
dataset17	Streptococcus_pneumoniae_TIGR4
dataset18	Streptococcus_sanguinis_SK36
dataset19	Mycobacterium_tuberculosis_H37Rv
dataset20	Mycoplasma_genitalium_G37
dataset21	Salmonella_enterica_serovar_Typhimurium_LT2
dataset22	Streptococcus_pneumoniae_R6
dataset23	Campylobacter_jejuni_NCTC_11168___ATCC_700819
dataset24	Salmonella_enterica_serovar_Typhimurium_14028S
dataset25	Vibrio_cholerae_O1_biovar_El_Tor_N16961

Geptop的改进

- 公式的改进

  由于最初的Geptop的公式可能不容易让人理解，所以决定将公式改的更容易理解一些。

- 乘法形式：

$$S_i = \prod_{j=1}^{N} \left({}^{M_{ij}} \big/ {}_{D_i} \right)$$

- 加法形式：

$$S_i = \sum_{j=1}^{N} \left({}^{M_{ij}} \big/ {}_{D_i} \right)$$

Geptop的改进

- 代码的改进

  之前的Geptop没用使用并行计算，不能充分发挥服务器的性能；而计算blast以及进化距离都会花费很长的时间，每次预测一个物种耗时很长，所以将代码改成并行计算。

- 并行模块的使用

  在python中，有一个简易的并行模块multiprocessing，我们使用的就是该模块中的Pool。然后，将Geptop中每个任务模块细分成多个小任务模块，使用Pool功能对这些小任务模块进行并行计算。

Geptop改进结果

- 通过跨物种交叉检验得出各个物种AUC值结果如下

		原始(19)	原始(25)	加法	乘法
DataSet1	Acinetobacter baylyi ADP1	0.8477586	0.8738874	0.8739901	0.8739912
DataSet2	Bacillus subtilis 168	0.951541	0.9539095	0.9535579	0.9535588
DataSet3	Caulobacter crescentus	0.8527213	0.8893486	0.8892929	0.8892932
DataSet4	Escherichia coli MG1655	0.9461658	0.9516198	0.9515552	0.9517142
DataSet5	Francisella novicida U112	0.8422643	0.8576055	0.8587216	0.8587216
DataSet6	Haemophilus influenzae Rd KW20	0.5668792	0.5714339	0.5678644	0.567866
DataSet7	Helicobacter pylori 26695	0.6080763	0.6224288	0.6219186	0.6219112
DataSet8	Mycobacterium tuberculosis H37Rv	0.7231095	0.7354756	0.7372401	0.7372472
DataSet9	Mycoplasma genitalium G37	0.7209813	0.7171358	0.7169994	0.7169994
DataSet10	Mycoplasma pulmonis UAB CTIP	0.8624043	0.8652132	0.86386	0.8638327
DataSet11	Pseudomonas aeruginosa UCBPP-PA14	0.7968078	0.8023147	0.8020563	0.8020574
DataSet12	Salmonella enterica serovar Typhi	0.9507245	0.960909	0.9643883	0.9646953
DataSet13	Salmonella typhimurium LT2	0.7030382	0.707959	0.7070655	0.7073407
DataSet14	Staphylococcus aureus N315	0.8329165	0.8301666	0.830221	0.8310539
DataSet15	Staphylococcus aureus NCTC 8325	0.880827	0.8834697	0.8815059	0.8831658
DataSet16	Streptococcus pneumoniae TIGR4	0.8107498	0.8152656	0.8112942	0.8139441
DataSet17	Streptococcus pneumoniae R6	0.7269089	0.7204354	0.7200556	0.7214918
DataSet18	Streptococcus sanguinis	0.9593236	0.9591403	0.958577	0.9585714
DataSet19	Vibrio cholerae N16961	0.6138469	0.6976393	0.7009057	0.7009071
mean		0.799844463	0.811334616	0.811108932	0.811492789
var		0.014647043	0.013592554	0.013684014	0.013689179
pec		0.978	0.9825246	0.982346	0.9825242

Geptop改进结果

- 改成并行计算过后，运行时间得到了极大的提升。以大肠杆菌为例，在改进之前，运行时间为6426秒；而改进过后，运行时间提高到了1572秒，提升了接近四倍。

Thank You