聚普体侧信息一体化软件 V1.0

（使用说明书）

目录

1. 引言
   1. 编写目的
   2. 背景
   3. 参考资料
2. 用途
   1. 功能
   2. 性能
3. 运行环境
   1. 硬件设备
   2. 支持软件
4. 使用过程
   1. kmash功能的使用说明
   2. GeneScan功能的使用说明

**1引言**

**1.1编写目的**

本文档首先介绍了genomediff，这是一种用于测量紧密相关的基因组之间的进化距离的软件工具。这些距离是Jukes-Cantor校正的散度在成对的基因组之间，即它们之间每个碱基的突变数。然后介绍了LTRharvest，这是一种从头预测在基因组序列中LTR逆转座子的软件工具。 LTRharvest计算潜在的LTR逆转座子的边界位置在基因组目标序列的持久索引结构上，增强后缀数组。这两种基因计算工具都是基因数据分析中常用的工具，对于取得供临床指导的指标有着广泛应用。

**1.2背景**

这一系统由昆山聚普生物信息技术有限公司开发，使用者是进行生物信息数据比对的有关分析人员。

**1.3参考资料**

**应用回归分析（R语言版）** [何晓群](https://book.jd.com/writer/%E4%BD%95%E6%99%93%E7%BE%A4_1.html) 著

陈铭，包家立，（2013）《生物信息学》（第二版）科学出版社

林标杨 ，（2012）《系统生物学》浙江大学出版社

**2用途**

**2.1功能**

genomediff是一种用于测量紧密相关的基因组之间的进化距离的软件工具。这些距离是Jukes-Cantor校正的散度在成对的基因组之间，即它们之间每个碱基的突变数，该距离称为Kr。计算所有成对的距离都是无对齐的，但是所产生的距离具有相同的生物学性意味着好像是通过多序列比对计算的。本软件能处理紧密相关的距离。genomediff是用C语言编写的，基于GenomeTools库。可以在不进行任何更改的情况下在32位和64位平台上编译源代码到源头。

LTRharvest是一种从头预测在基因组序列中LTR逆转座子的软件工具，计算潜在的LTR逆转座子的边界位置在基因组目标序列的持久索引结构上，增强后缀数组。基因组目标序列是FASTA格式的完整染色体DNA序列。但是每个FASTA格式的DNA序列都可以传递给软件。为了进行预测，LTRharvest实现了几个过滤器。这些被连续应用于序列数据以拒绝不符合序列，长度或距离特征的候选对象LTR逆转座子。由于这些功能大部分是特定于物种的，因此可以切换每个过滤器开启或关闭，并且对于某个LTR反转录转座子模型参数化。LTRharvest用C编写，基于GenomeTools库。 LTRharvest被单个名为gt的二进制文件调用。 LTRharvest在增强的后缀数组索引上运行，该索引已存储在文件上。该索引需要由程序后缀程序构造，该程序后缀程序也是GenomeTools二进制gt。

**3运行环境**

**3.1硬件设备**

PC机要求 ：x86通用计算机系统，4核cpu，8G内存，1T硬盘

打印机 ： Windows支持的打印机

**3.2支持软件**

windows 10 或以上操作系统

开发语言 ：c#，python

**4使用过程**

**4.1** genomediff**命令行选项：**

由於基因組是GenomeTools的一部分，因此它的調用方式如下：

gt genomediff [options] (INDEX | -indexname NAME SEQFILE SEQFILE [...])

其中INDEX是不包含包含基因組的編碼序列延伸的路徑進行比較，NAME是要根據給定構建的編碼序列的名稱序列文件。表1給出了所有可能選項的簡短描述。

清單1：示例單位文件：“ units”部分是必填項，“ genome1 / 2”部分是名稱示例，

文件名是在命令行上或在索引構建過程中指定的路徑。

units = {

genome1 = { "file1.fas", "file2.fas" },

genome2 = { "path/file3.fas", "file4.fas" }

}

* **输入文件：**

基因组工具可以处理三种类型的准备索引。第一个是编码的序列，可以由encseq准备。给定一个编码序列，基因组将在内存中构建增强的su x数组，并使用该索引计算Kr。第二个是由工具su xerator（请参阅gt suffixerator -help）准备的增强的su x数组，第三个使用packedindex工具构建的压缩FM索引（请参见gtpackedindex mkindex -help。不建议使用FM索引，因为Kr的计算非常重要更长。

输入选项

-indextype type指定索引的类型，其中之一：esajpckjencseq。其中encseq是

编码序列和增强的su x阵列将被构建

仅在内存中。默认值：encseq

-unitfile lename指定基因组单位，请参见下面的描述。默认值：unde-

内德

输出选项

-indexname name要构造的encseq的基本名称。默认值：未定义

ESA选项

-mirrored虚拟附加每个序列默认值的反补：

不

-pl n为存储桶排序建议指定pre x长度：不使用

争论;那么合理的pre x长度会自动确定-

挖出来的默认值：0

-dc n指定差异覆盖值。默认值：0

-memlimit n指定在索引连接期间要使用的最大内存量

指令（以字节为单位，允许使用关键字“ MB”和“ GB”）。去

故障：未定义

杂项选项

-v冗长。默认值：否

-help显示基本选项的帮助并退出。

-help +显示所有选项的帮助并退出。

-version显示版本信息并退出。

另一种方法是直接指定序列文件的名称。选项--indexname是必需的在这种情况下。给定名称将用于在磁盘上存储编码序列。文件格式可以是GenomeTools支持的任何序列格式。无论哪种方式，每个给定的序列le都将被视为一个基因组单位，无论该文件内的序列数。为基因组单位赋予名称以外的其他名称，或将les组合为单个基因组一个单位可以用--unitfile选项给出一个单位文件。显示了示例文件的格式在清单1中。

**输出**

标准输出流上的输出由一行与基因组数量或比较的单位。随后是成对距离的二次矩阵，其中每个行由文件名或单元名以及制表符分隔的距离值组成。根据gt调用的选项，可能会有其他输出，其中每行是在标准错误流上以'＃'开头，并以'debug：'开头的附加输出。

* **例子**

本节描述了两个示例场景，第一个是多个基因组的比较组织在多个独立的FASTALES中，第二个是两个基因组的比较每个由多个文件组成。

**比较单独文件中的基因组**

考虑三个les基因组1.fas，基因组2.fas和基因组3.fas，每个都可能包含

多个FASTA条目。我们的机器有2个GiB RAM。假设索引构造会需要5 GiB，我们需要将其分成至少三个相等大小的部分，或者限制最大内存要求。

gt genomediff -indexname 3 genomes \

-memlimit 1500 MB \

genome1 .fas genome2 . fas genome3 .fas

--memlimit应该合理地小于可用的主内存。

这将在终端上输出距离矩阵，并使用

当前目录中的basename 3genomes。

为了将结果保存到文件中，请使用终端重定向：gt基因组差异...>外文件。

le outfile可能看起来像这样：

genome1.fas 0.000000 0.115125 0.267473

genome2.fas 0.115125 0.000000 0.293082

genome3.fas 0.267473 0.293082 0.000000

此制表符分隔的表格可与Phylip或R一起使用，以计算出

遗传树。

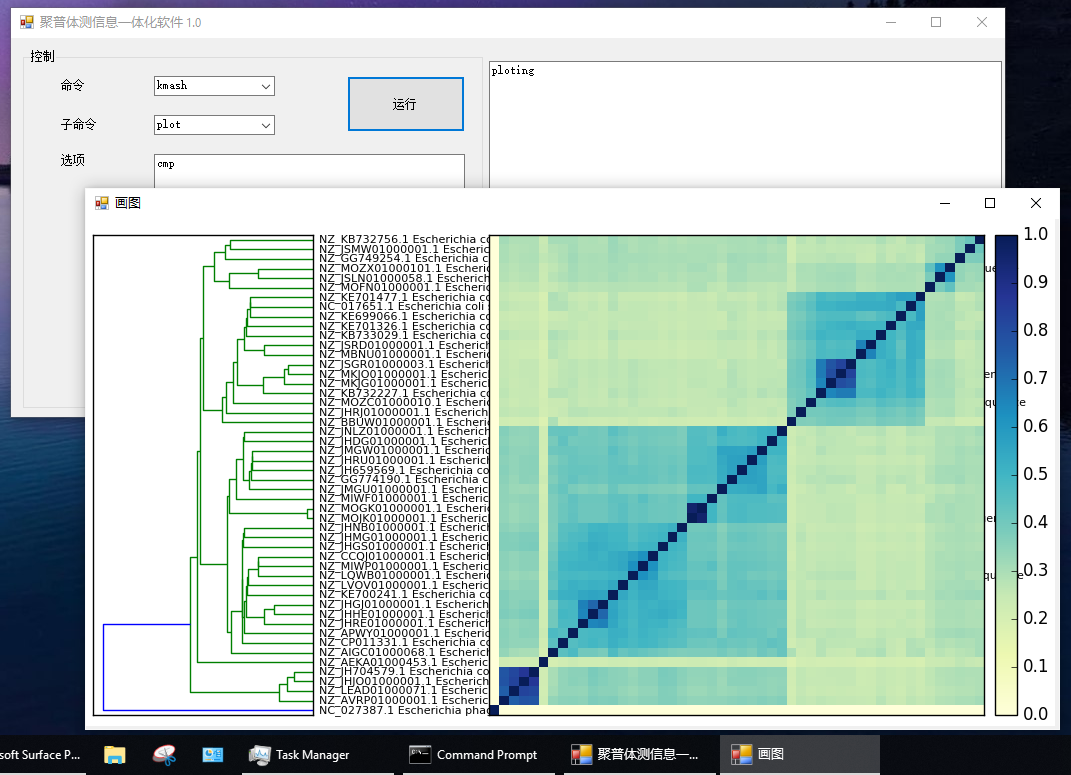
如果给定文件的增强的su x数组具有

名称3genomes idx已经存在于磁盘上，如下所示：

gt genomediff -indextype esa 3 genomes\_idx > outfile

要重用现有的编码序列，只需提供其基本名称：

gt genomediff 3 genomes > outfile



比较多个les中的两个基因组

假设我们有两个基因组，它们由位于不同文件中的多个染色体组成。为了

例如，基因组1由g1 chr1.fas和g1 chr2.fas组成，而基因组2的两个les是

相应地命名。 le可以这样组织：

units = {

genome1 = { "g1\_chr1 .fas", "g1\_chr2 .fas" },

genome2 = { "g2\_chr1 .fas", "g2\_chr2 .fas" }

}

在我们的示例中，单位文件的名称将为单位。

现在我们可以这样称呼基因组：

gt genomediff -indexname 2 genomes \

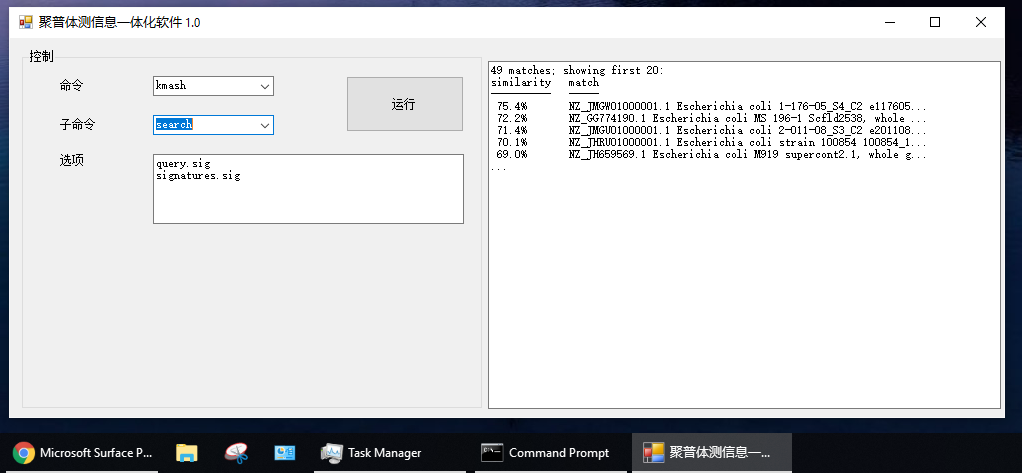
-unitfile units \

g1\_chr1 .fas g1\_chr2 . fas g2\_chr1 .fas g2. chr2 . fas > output

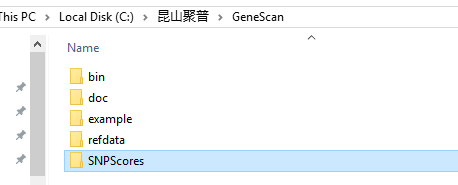
文件输出如下所示：

genome1 0.000000 0.115125

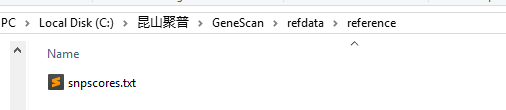
genome2 0.115125 0.000000

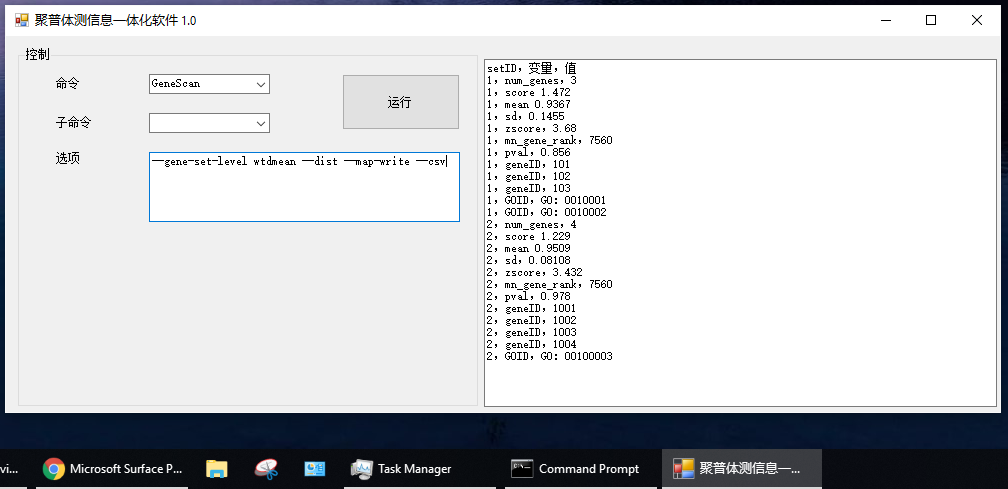


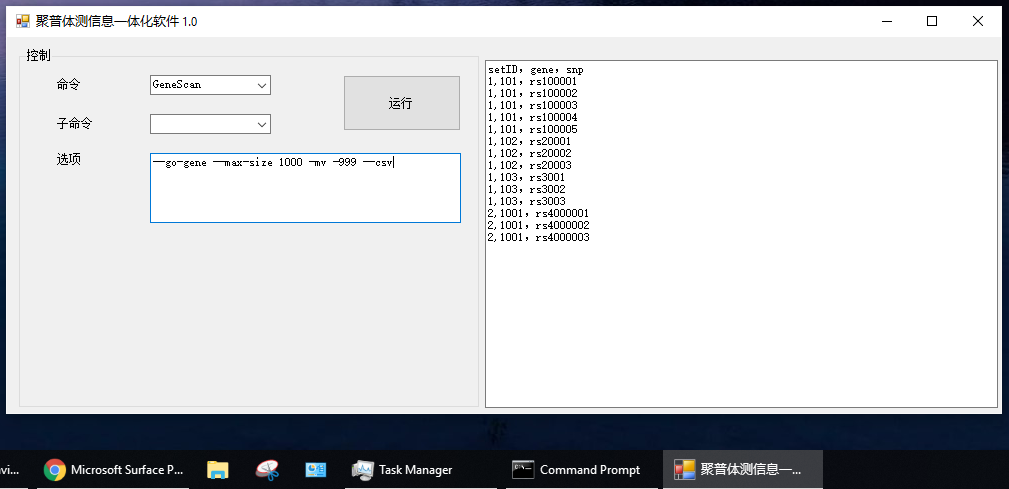
* **文件夹结构**



* **输入文件**







用法

根据以下规则，某些文本将使用不同的字体突出显示。

打字机字体用于软件工具的名称。

小型打字机字体用于文件名。

程序选项使用脚注大小的带有'-'开头的打字机字体。

小斜体字体用于选项的参数。

LTRharvest的选项

由于LTRharvest是gt的一部分，因此LTRharvest的名称如下。

gt ltrharvest -index indexname [选项]

其中indexname表示由

程序Suffixerator。所有可能选项的概述，每个选项都有简短的一行描述

表1中给出了option选项。所有选项只能指定一次。

输入选项

-index索引名

指定增强后缀数组索引的名称。索引必须包含表suf，

lcp，des和tis。如何使用

程序Suffixerator可以在第3节中找到。

表1：按类别分类的LTRharvest选项概述。

输入选项

-index指定增强后缀数组索引的名称

-range指定要搜索LTR反转录转座子候选序列的输入序列中的范围

输出选项

-out为多种FASTA格式的预测指定输出文件名

-gff3为GFF3格式的预测指定输出文件名

-outinner为预测内部区域的多个FASTA文件指定outputfilename

筛选选项

-seed指定最小的种子长度，以进行精确的最大重复

-minlenltr指定每个LTR的最小长度

-maxlenltr指定每个LTR的最大长度

-mindistltr指定LTR起始位置的最小距离

-maxdistltr指定LTR起始位置的最大距离

-like指定相似度阈值，范围为[1..100％]

-mintsd指定每个TSD的最小长度

-maxtsd指定每个TSD的最大长度（需要-mintsd）

-motif指定由2个核苷酸startmotif + 2个核苷酸endmotif组成的回文基序，

-motifmis指定主题中最大不匹配数（需要-motif）

-vic指定核苷酸数（在左侧和右侧）

选项-index是必需的。

范围x y

在输入序列中指定要搜索LTR反转录转座子候选物的范围。

也就是说，如果x = 1000和y = 10000，则仅当候选人在排名之后开始时才被报告

1000，并且在各自的序列坐标中位于位置10000之前。

如果x = y = 0（默认值），则搜索整个序列。

2.3输出选项

结果以表格形式在stdout上报告，并且可以使用表示法轻松地写入文件

> resultfile，如下所示：

gt ltrharvest -index indexname [选项]>结果文件

此外，以下选项将额外的信息写入文件

输出文件

指定将要写入预测的文件的名称。每个预测将是

由单个FASTA条目表示。

-内部输出文件

指定内部区域（无侧翼的预测顺序）的文件名

LTR序列）将被写入。每个预测将由一个单独的FASTA表示

入口。

-gff3输出文件

指定将要写入预测的文件的名称。每个预测将是

由单个GFF3 [2]条目表示。

2.4过滤器选项

过滤器选项提供了排除不需要的序列，长度或

距离功能。如果用户未设置特定选项，则该选项的默认值为

设置，但选项-mintsd，-maxtsd和-motif除外。因此，如果未设置任何过滤器选项

用户将进行LTR逆转座子的预测，而无需搜索目标位点

重复（TSD）或特定的LTR起始-终止基序。

种子莱克斯

指定确切最大重复次数的最小长度。只有那些与指定重复

在寻找候选对的过程中分析最小长度。精确最大

长度低于该阈值的重复重复将不再考虑。 Lex必须是一个积极的人

整数。如果用户未选择此选项，则默认情况下Lex设置为30。

-minlenltr Lmin

指定每个LTR的最小长度。 Lmin必须指定为正整数。如果这

用户未选择“选项”，则Lmin默认设置为100。

-最大

指定每个LTR的最大长度。 Lmax必须指定为正整数。如果这

用户未选择该选项，则默认情况下Lmax设置为1000。

-mindistltr Dmin

指定LTR起始位置的最小距离。必须将Dmin指定为正数

整数。如果用户未选择此选项，则默认情况下Dmin设置为1000。

-maxdistltr Dmax

指定LTR起始位置的最大距离。必须将Dmax指定为正数

整数。如果用户未选择此选项，则默认情况下Dmax设置为15000。

-相似相似度阈值

指定两个LTR之间的最小相似度值。论点相似度阈值

必须从[0,100]范围内选择，并且表示百分比。如果未选择此选项

用户将默认值设置为85％。

-mintsd TSDminlen

如果选择此选项，将搜索目标站点重复项（TSD）。

TSDminlen的最小TSD长度。如果用户未选择此选项，则搜索

将执行最小TSD长度为4的TSD。如果设置了此选项，但没有最大值

TSD长度由选项-maxtsd指定，然后将最大TSD长度设置为20，

默认。

-maxtsd TSDmaxlen

该选项需要选项-mintsd。如果选择此选项，则搜索目标站点

重复（TSD）将以最大TSD长度TSDmaxlen进行。

-motif expr

在起始处指定2个核苷酸作为起始基序，在末端指定2个核苷酸作为终止基序

和每个LTR的结尾。仅回文基序序列-

基序序列等于其互补序列向后读取-允许，例如tgca。

键入核苷酸，没有任何空格将它们分开。如果没有选择此选项

用户，则将不检查候选对是否包含主题。如果设置了此选项，但是

-motifmis选项未指定允许的不匹配数，然后搜索

将会进行精确的图案设计。

-基序

该选项需要选项-motif。指定允许的不匹配数

论据如果未设置此选项，则将搜索确切的图案。这

非负整数n必须从范围[0; 3]。

-vic l

指定左侧和右侧（附近）的核苷酸位置l的数目，这将

在LTR的5'和3'预测边界附近搜索TSD和/或一个图案

反转录转座子。此选项只有一个作用，如果选择-mintsd和/或选项-motif是

切换到。如果用户未选择此选项，则l的默认值为60。

-重叠nojbestjall

指定有关嵌套和/或重叠的LTR反转录转座子预测的输出。如果

选择参数no，则不会在其中报告嵌套或重叠的预测

输出。如果选择了最佳参数，则在两个或多个嵌套或重叠的情况下

预测，仅是其间具有最高相似性的LTR反转录转座子预测

LTR将被报告。如果选择所有参数，则所有LTR反转录转座子预测

将报告是否存在嵌套和/或重叠的预测。如果这个选项是

如果未由用户选择，则默认设置参数为best的选项。

2.5对齐选项

X-drop扩展过程允许搜索退化的LTR。对齐选项提供

有机会控制最大重复种子的这种延伸。如果有特定的对齐方式

选项不是由用户设置的，此选项的默认值将被设置。

-xdrop X

指定xdrop值X，以在两个方向上扩展种子重复，以实现匹配，

不匹配，插入和删除。参数X必须为正整数或0。

只要涉及匹配，不匹配，插入，

并且删除的分数小于T X，其中T表示看到的最大分数，因此

远的。如果用户未选择此选项，则默认情况下X设置为5。

垫得分

为X-drop扩展过程指定正匹配分数。如果未选择该选项

由用户设置，默认值为2。

-mis分数

为X-drop扩展过程指定负的不匹配得分。如果不是这个选项

由用户选择，默认值为 2。

-ins得分

为X-drop扩展过程指定否定插入得分。如果不是这个选项

由用户选择，默认值为 3。

-del分数

为X-drop扩展过程指定否定的删除分数。如果不是这个选项

由用户选择，默认值为 3。

2.6其他选项

-v

此选项启用详细模式。这意味着，有关

在运行期间，处理将被打印到标准输出。这包括一长串已开启的清单

或关闭选项。

长输出

此选项还会将有关检测到的TSD和/或主题的信息打印到stdout，

如果用户已经选择了对TSD和/或主题的搜索。此选项需要

选项-mintsd和/或-motif。

-帮助

LTRharvest将在stdout上显示所有选项的摘要，并以退出代码0终止。

例子

在本节中，将介绍LTRharvest的示例应用程序。在小节3.1中，

给出了使用LTRharvest的不同选项的信息。然后，第3.2小节给出了预测的示例

LTR逆转座子在整个酿酒酵母基因组上的表达在3.3小节中，

显示了LTRharvest输出的聚类过程。请注意，此步骤不属于

LTRharvest由Vmatch进行[7]。

3.1使用LTRharvest的不同选项

作为目标序列文件，我们使用一些包含酿酒酵母的FASTA文件chr02.19970727.fsa.gz

基因组序列，染色体2。请注意，该文件是gzip格式的压缩文件（因为

.gz的结尾）。此格式可以由程序Suffixerator处理。

首先，我们创建增强的后缀数组。我们使用选项-tis，-suf，

-lcp，-des，-ssp和-sds，因为LTRharvest需要相应的表。此外，我们指定

-dna，因为我们处理DNA序列。

$ gt suffixerator -db chr02.19970727.fsa.gz -indexname chr02.19970727.fsa -tis

-suf -lcp -des -ssp -sds -dna

# dna=yes

# indexname="chr02.19970727.fsa"

.

.

.

# TIME overall 1.37

现在我们可以将索引用于LTRharvest。第一个示例调用将仅使用默认参数

用于过滤器和对齐选项，而无需搜索TSD或LTR起始-结束基序。

$ gt ltrharvest -index chr02.19970727.fsa

# args=-index chr02.19970727.fsa

# predictions are reported in the following way

# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR)

sim(LTRs) seq-nr

# where:

# s = starting position

# e = ending position

# l = length

# ret = LTR-retrotransposon

6

# lLTR = left LTR

# rLTR = right LTR

# sim = similarity

# seq-nr = sequence number

259532 265448 5917 259532 259863 332 265117 265448 332 99.40 0

427672 430021 2350 427672 428170 499 429522 430021 500 91.60 0

29632 35597 5966 29632 29969 338 35259 35597 339 98.82 0

每条注释行均以注释符号＃开头。每个非注释行表示一个LTR反转录转座子

整个LTR反转录转座子起始和终止位置的预测，左侧

LTR实例和正确的LTR实例。此外，对于这些元素中的每一个，

报告了相应的元素长度以及两个LTR的相似度百分比。最后

每行整数表示输入序列号，LTR反转录转座子预测

发生在。输入序列号从0开始计数。

使用可选参数-v调用LTRharvest可获得有关启用和禁用的更多信息

选项以及有关增强后缀数组索引和时间/空间的其他信息

消耗。此外，将选项-out和-outinner指定为运行结果的两个倍数

FASTA文件。

$ gt ltrharvest -index chr02.19970727.fsa -v -out pred-chr02.fsa

-outinner pred-inner-chr02.fsa

# args=-index chr02.19970727.fsa -v -out pred-chr02.fsa -outinner

pred-inner-chr02.fsa

# user defined options and values:

# verbosemode: On

# indexname: chr02.19970727.fsa

# outputfile: pred-chr02.fsa

# outputfile inner region: pred-inner-chr02.fsa

# xdropbelowscore: 5

# similaritythreshold: 85.00

# minseedlength: 30

# matchscore: 2

# mismatchscore: -2

# insertionscore: -3

# deletionscore: -3

# minLTRlength: 100

# maxLTRlength: 1000

# minLTRdistance: 1000

# maxLTRdistance: 15000

# overlaps: best

# minTSDlength: 0

# maxTSDlength: 20

# palindromic motif:

# motifmismatchesallowed: 4

# vicinity: 60 nt

# predictions are reported in the following way

# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR) sim(LTRs)

seq-nr

# where:

# s = starting position

# e = ending position

# l = length

7

# ret = LTR-retrotransposon

# lLTR = left LTR

# rLTR = right LTR

# sim = similarity

# seq-nr = sequence number

259532 265448 5917 259532 259863 332 265117 265448 332 99.40 0

427672 430021 2350 427672 428170 499 429522 430021 500 91.60 0

29632 35597 5966 29632 29969 338 35259 35597 339 98.82 0

220989 226919 5931 220989 221339 351 226574 226919 346 97.15 0

我们还使用-mintsd，-maxtsd，

-motif和-motifmis。

$ gt ltrharvest -index chr02.19970727.fsa -mintsd 5 -maxtsd 20 -motif tgca

-motifmis 0

# args=-index chr02.19970727.fsa -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0

# predictions are reported in the following way

# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR) sim(LTRs)

seq-nr

# where:

# s = starting position

# e = ending position

# l = length

# ret = LTR-retrotransposon

# lLTR = left LTR

# rLTR = right LTR

# sim = similarity

# seq-nr = sequence number

259532 265448 5917 259532 259863 332 265117 265448 332 99.40 0

29632 35590 5959 29632 29963 332 35259 35590 332 99.70 0

220996 226911 5916 220996 221329 334 226575 226911 337 97.33 0

最后，如果我们对TSD的顺序和长度以及

主题，我们选择-longoutput选项。这也是指定所有过滤器和

用户的对齐选项。

$ gt ltrharvest -index chr02.19970727.fsa -seed 30 -xdrop 5 -mat 2 -mis -2 -ins -3

-del -3 -minlenltr 100 -maxlenltr 1000 -mindistltr 1000 -maxdistltr 15000

-similar 90.0 -overlaps all -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0 -vic 60

-longoutput

# args=-index chr02.19970727.fsa -seed 30 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3

-minlenltr 100 -maxlenltr 1000 -mindistltr 1000 -maxdistltr 15000 -similar 90.0

-overlaps all -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0 -vic 60 -longoutput

# predictions are reported in the following way

# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) TSD l(TSD) m(lLTR) s(rLTR)

e(rLTR) l(rLTR) TSD l(TSD) m(rLTR) sim(LTRs) seq-nr

# where:

# s = starting position

# e = ending position

# l = length

# m = motif

# ret = LTR-retrotransposon

# lLTR = left LTR

8

# rLTR = right LTR

# TSD = target site duplication

# sim = similarity

# seq-nr = sequence number

259532 265448 5917 259532 259863 332 gtaat 5 tg..ca 265117 265448 332

gtaat 5 tg..ca 99.40 0

29632 35590 5959 29632 29963 332 ataat 5 tg..ca 35259 35590 332

ataat 5 tg..ca 99.70 0

220996 226911 5916 220996 221329 334 ggaat 5 tg..ca 226575 226911 337

ggaat 5 tg..ca 97.33 0

对整个酿酒酵母基因组的预测

作为目标序列文件，我们使用多个FASTA文件chrAll.19971001.fsa.gz，其中包含

1997年10月1日之前释放的所有16个染色体序列，可能由Kim等人使用。

在他们对逆转座子的全面调查中[6]。

首先，我们创建增强的后缀数组。我们使用选项-tis，-suf，

-lcp，-des，因为LTRharvest需要相应的表。此外，我们指定选项-dna，

因为我们正在处理DNA序列。

$ gt suffixerator -db chrAll.19971001.fsa.gz -indexname chrAll.19971001.fsa -tis

-suf -lcp -des -sds -dna

# dna=yes

# indexname="chrAll.19971001.fsa"

.

.

.

# TIME overall 108.35

现在我们可以将索引与LTRharvest一起使用。除了过滤器和对齐选项之外，我们还选择

option -out，用于将预测的LTR反转录转座子序列打印到文件中。

$ gt ltrharvest -index chrAll.19971001.fsa -seed 100 -minlenltr 100 -maxlenltr 1000

-mindistltr 1000 -maxdistltr 15000 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3

-similar 90.0 -overlaps best -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0

-vic 60 -longoutput

-out pred-chrAll.fsa

# args=-index chrAll.19971001.fsa -seed 100 -minlenltr 100 -maxlenltr 1000

-mindistltr 1000 -maxdistltr 15000 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3

-similar 90.0 -overlaps best -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0

-vic 60 -longoutput -out pred-chrAll.fsa

# predictions are reported in the following way

# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) TSD l(TSD) m(lLTR) s(rLTR) e(rLTR)

l(rLTR) TSD l(TSD) m(rLTR) sim(LTRs) seq-nr

# where: 160239 166163 5925 160239 160575 337 ggttc 5 tg..ca 165827 166163 337

ggttc 5 tg..ca 100.00 0

29632 35590 5959 29632 29963 332 ataat 5 tg..ca 35259 35590 332

ataat 5 tg..ca 99.70 1

.

.

9

.

844407 850335 5929 844407 844744 338 gaaat 5 tg..ca 849998 850335 338

gaaat 5 tg..ca 100.00 15

56452 62375 5924 56452 56788 337 gttat 5 tg..ca 62039 62375 337

gttat 5 tg..ca 100.00 15

LTRharvest输出的序列聚类（可选）

除了LTRharvest进行的预测过程外，还对预测序列进行了聚类分析

被推荐。在这里，我们从Vmatch中选择单连锁聚类分析程序

软件工具[7]（不是GenomeTools二进制gt的一部分），以显示此任务的执行方式

可以完成。程序需要根据预测的序列构造索引

mkvtree是Vmatch的一部分。

$ mkvtree -db pred-chrAll.fsa -dna -pl -allout -v

reading file "pred-chrAll.fsa"

...

.

.

.

overall space peak: main=2.61 MB (10.01 bytes/symbol), secondary=0.53 MB

现在，我们可以使用vmatch（它是Vmatch的一部分）对预测序列进行聚类。

Tab中给出了合理的参数集。 2，第11页。有关详细信息，请参见Vmatch手册[7]。

选项说明。以下命令计算所有（数据库）序列的集群

在pred-chrAll.fsa中。每个簇都有一个唯一的簇号，后跟序列号

包含在群集中。

$ vmatch -dbcluster 95 7 Cluster-pred-chrAll -p -d -seedlength 50

-l 1101 -exdrop 9 pred-chrAll.fsa

# args=-dbcluster 95 7 Cluster-pred-chrAll -p -d -seedlength 50

-l 1101 -exdrop 9 pred-chrAll.fsa

# 3 clusters

# 45 elements out of 46 (97.83%) are in clusters

# 1 elements out of 46 (2.17%) are singlets

# 1 cluster of size 2

# 1 cluster of size 3

# 1 cluster of size 40

0: 32 41 42 36 39 13 40 43 37 30 44 38 18 21 17 34 35 31 12 20

33 23 26 27 6 28 3 29 25 1 24 9 19 2 7 22 4 10 5 0

1: 8 15 11

2: 14 45

220989 226919 5931 220989 221339 351 226574 226919 346 97.15 0

将在5'和3'边界附近搜索TSD和/或图案

-overlaps指定nojbestjall

对齐方式

-xdrop为种子扩展指定xdrop值

-mat为种子扩展指定matchscore

-mis为种子扩展指定mismatchscore

-ins为插入扩展指定insertscore

-del为种子扩展指定deletescore

杂项选项

-v详细模式

-longoutput附加主题/ TSD输出（需要-mintsd或-motif）