1. 引言

1.1 编写目的

拷贝数变异（Copy Number Variation, CNV）是指基因组广泛存在的 50 bp 到数百万 bp 的片段重复或者缺失。研究表明，不少人类复杂疾病，动植物的重要经济性状都和拷贝数变异有密切联系。随着高通量测序技术的发展及测序费用逐步降低，基于大规模群体重测序方法来解读动植物重要经济性状的分子遗传机制的研究越来越多。为了在群体水平快速准确检测 CNV，本研究开发了 CNVcaller。

1.2 项目背景

大规模群体重测序数据可用于群体水平 CNV 区域的检测，并进一步通过全基因组关联分析、遗传进化分析寻找与重要经济性状相关的功能变异。但是大群体 CNV 检测大大增加了对计算资源的需求。另外，与人类参考基因组组装质量相比，目前动植物基因组存在高比例的 gap，unplaced scaffolds 和错误组装的片段重复等，严重影响了 CNV 检测的准确性。因此，需要快速，简便，适应性广的群体 CNV 检测软件。

1.3 术语缩写词

CNV（Copy Number Variation）：拷贝数变异，指基因组上存在的 50 bp 到数百万 bp 的片段重复或者缺失。

CNVR（Copy Number Variation Region）：拷贝数变异区域，指将源于不同个体的 CNV 合并成具有统一边界的区域，该区域称为拷贝数变异区域。

RD（read depth）：读段深度，指基因组上某个特定区域比对上的测序读段数目。理论上重复区域的 RD 高于正常区域，缺失区域的 RD 低于正常区域。

VCF（Variant Call Format）：记录基因组遗传变异的常用文件格式。

BAM（Binary Alignment Map）：高通量测序得到的短序列通过比对工具（如 BWA）比对到参考基因组后得到的二进制文件，主要记录了短序列的比对信息，包括比对位置，比对质量等。

gap：参考基因组中没有组装出来的部分，一般用 N 填充。

1.4 参考资料

2. 软件概述

2.1 目标

为了快速准确检测大规模动植物群体基因组拷贝数变异。

2.2 功能

CNVcaller 主要用于动植物全基因组拷贝数变异的检测。其主要步骤为：1.将参考基因组分割成指定大小的窗口；2.统计窗口中匹配上的读段数；3.根据窗口之间的相似度进行绝对拷贝数校正；4.对校正后的窗口读段数进行 GC 含量校正和标准化；5.利用标准化的读段信号在群体中检测拷贝数变异并进行基因型判定。

2.3 性能

输入

输入数据为 BWA 比对后的 BAM 文件，由于 CNVcaller 会对基因组上高相似度（默认大于 97%）的区域做绝对拷贝数校正，所以**不推荐在比对完成后进行比对质量的过滤**。此外，推荐只保留测序深度在 5X 以上的样本。

输出

输出结果为 VCF 格式，可直接用于后续全基因组关联分析和进化选择分析。

准确性

利用绵羊 3 个家系重测序数据（测序深度 10X），通过孟德尔遗传错误检测发现，CNVcaller 在片段重复区域的 FDR 为 4%，片段缺失区域的 FDR 为 2%。

计算效率

经测试，CNVcaller 仅用一个计算节点，在两天内完成 200 个家畜（测序深度为 10X）的拷

贝数变异检测。预计可在一周内报告 1,000 个哺乳动物的 CNVRs。此外，CNVcaller 运行效率不受基因组复杂度的影响，如包含大量 scaffolds 的基因组序列草图、小麦基因组和泛基因组。

运行环境

程序运行需要 Linux 环境，并预安装以下软件：

Perl5+：已通过 5.10.1 版本的测试

Samtools：已通过 1.3 版本的测试

Python：已通过 3.6 版本的测试

安装说明及示例下载

若服务器联网，可在终端通过以下命令 git clone https://github.com/JiangYuLab/CNVcaller.git 直接进行安装；若服务器不能联网，可通过本地下载后再上传到服务器进行安装。

为了方便大家能够快速入手 CNVcaller 的计算流程，大家可以从 ftp://jiang:jiang@animal.nwsuaf.edu.cn/CNVcaller/demo/ 下载测试数据进行练习。

求助方式

安装、运行遇到任何问题，或者发现任何程序漏洞可以发送邮件至 yu.jiang@nwafu.edu.cn，

或登录 github（https://github.com/JiangYuLab/CNVcaller）留言。

软件主页

http://animal.nwsuaf.edu.cn/software

https://github.com/JiangYuLab/CNVcaller

程序说明及示例

CNVcaller 包含了四步：一个 Perl 脚本，两个 shell 脚本和一个 Python 脚本。在运行前，需

要基于 CNVcaller 的安装环境修改两个 shell 程序中 CNVcaller 的安装路径，默认是当前工作目录，如下图所示。

```
#!/bin/sh
# CNVcaller installation directory
export CNVcaller=`pwd`
echo "CNVcaller install directory $CNVcaller"
```

运行示例

本部分将通过例子文件对 CNVcaller 每一步运行方法，对应参数和输出进行说明。

构建参考基因组数据库

```
perl CNVReferenceDB.pl

Program: Divide the genome into fixed-sized (window_size (bp)) windows and generate th
e table file containing information about the count of GC, repeat (derived from repeat
mask results) and gap

Usage: CNVReferenceDB.pl <ref>

  -w  window size (bp)
      default = 800

  -l  minimum GC content to include a window in reference database
      default = 0.2

  -u  maximum GC content to include a window in reference database
      default = 0.7

  -g  maximum gap content to include a window in reference database
      default = 0.5
```

作用：将参考基因组按用户指定大小（-w）的滑动窗口及一定的步长（内置是滑动窗口大小的一半）分别统计基因组上每个窗口的 GC、repeat 及 gap 含量。

输入文件：参考基因组（reference.fa）文件

运行例子：按照 400 bp 的滑动窗口，200 bp 的步长生成参考基因组（reference.fa）的数据库。

perl CNVReferenceDB.pl reference.fa -w 400

输出文件：上述运行完成会在工作目录下生成一个名为“referenceDB.400”的文件，其中的“400”为滑动窗口的大小，文件的每列信息如下：

第一列，染色体名称

第二列，窗口在对应染色体上的序号

第三列，窗口实际起始位置

第四列，GC 含量

第五列，重复序列含量

第六列，gap 比例

参数详解：

-w 滑动窗口大小（bp）。默认 800。注：对于≥10X 的测序数据，我们推荐用 400-1,000 bp 的窗口大小，对于<10X 的测序数据我们推荐用 1,000-2,000 bp 的窗口大小。增大窗口可以降低假阳性率，但同时会增加假阴性率。

-l 窗口可含有的最低 GC 比例，低于该比例的窗口将不会进入后续计算。默认 0.2。

-u 窗口可含有的最高 GC 比例，高于该比例的窗口将不会进入后续计算。默认 0.7。注：一般动植物基因组中 GC 比例低于 20%和高于 70%的窗口数量较少，而且目前二代测序较难测到这些区域，故这类窗口不建议进入统计。

-g 窗口可含有的最高 gap 比例，高于该比例的窗口将不会进入后续计算。默认 0.5。注：当一个窗口存在任意比例的 gap，都意味着 gap 两侧的序列可能存在组装错误，所以建议不对这类窗口进行统计计算。如果参考基因组 gap 数量过多（如 N50 小于 20Kb，也就是大约每 20 Kb 会有一个 gap），导致不进入统计的窗口数超过全基因组的 5%，可以适当将 gap 比例参数设置为 0.1。

计算每个窗口的绝对拷贝数

```
bash Individual.Process.sh
CNVcaller install directory /stor9000/apps/users/NWSUAF/2015060152/script/CNVcaller/GitHub/04.version
Usage: Individual.Process.sh -b <BAM> -h <header> -d <dup> -s <sex_chromosome>
required options:
-b|--bam          alignment file in BAM format
-h|--header       header of BAM file, the prefix of output file [same with SM tag of input BAM file]
-d|--dup          duplicated window record file used for absolute copy number correction
-s|--sex          the name of sex chromosome
```

作用：计算每个个体基因组所有窗口的绝对拷贝数。该过程主要包括三步：1. 解析每个样本的 BAM（BWA 比对生成）文件，并统计每个窗口的读段数；2. 对高相似度的（≥97%）窗口的读段数进行合并；3. 对每个窗口合并后的读段数进行 GC 偏好性校正并将其除以所有窗口校正后读段数的中位数，以获得该窗口的绝对拷贝数。在这一步，每个样本运行需要约 500 MB 内存，所以一个节点可以同时提交多个任务。

输入：上一步输出的数据库文件“ReferenceDB.400”和 BAM 文件。默认数据库文件放在当前目录下（用户可自己修改 shell 程序）。

输出：运行结束后，三个默认文件夹（RD_raw、RD_absolute、RD_normalized）将会在当前目录下被创建，分别包含了每个样本的全基因组所有窗口原始读段数、通过 link 文件合并后的读段数，以及每个样本经 GC 测序偏斜校正和标准化后的绝对拷贝数。标准化的文件名显示了该个体基因组所有窗口的读段数平均值、标准差与性别（1 为 XX 或 ZZ，2 为 XY 或 ZW），其中平均值和标准差可用于样本的质控。绝对拷贝数为 1 时，表示为正常的拷贝数，即正常二倍体；0.5 表示杂合缺失；0 表示纯合缺失；1.5 表示杂合重复；2 表示纯合重复；绝对拷贝数超过 2 表示复杂的多次重复。

运行例子：统计 ERR340328 个体每个窗口比对上的读段数，并通过 link 文件合并高相似度窗口的读段数，最后进行 GC 校正和绝对拷贝数的标准化，其中记录基因组高相似度窗口的文件是“link”，性染色体的名称是“X”。

```
bash Individual.Process.sh -b ERR340328.bam -h ERR340328 -d link -s X
```

参数详解：

-b BWA 比对生成的 BAM 文件。

-h BAM 文件的标头信息，需要与 BAM 文件中 SM 标签保持一致（用户可以通过 samtools view -H BAM 查看）。

-d 校正所需要的 link 文件。其中，常用动植物 800 bp 窗口的 link 文件可通过 ftp://jiang:jiang@animal.nwsuaf.edu.cn/CNVcaller/database/ 直接下载。如果该站点不包含所需物种的 link 文件，可由用户自行生成（详细信息请见后续）。

-s 性染色体名称。CNVcaller 会根据给定的性染色体所有窗口读段数的中位数与所有常染色体窗口读段数的中位数比例来确定该个体的性别。当指定性染色体窗口读段数中的位数大于常染色体窗口读段数中位数的 75%且小于 150%时，该个体判定为 XX/ZZ（即标准化后的文件名性别记录为 1）；当指定性染色体窗口读段数的中位数大于常染色体窗口读段数的中位数的 25%且小于 75%时，该个体判定为 XY/ZW（即标准化后的文件名性别记录为 2）。**对于没有性染色体的物种，-s 参数可以设置为“none”**。

拷贝数变异区域的确定

```
bash CNV.Discovery.sh
CNVcaller install directory /stor9000/apps/users/NWSUAF/2015060152/script/CNVcaller/GitHub/04.version
Usage: CNV.Discovery.sh -l <normalizedFileList> -e <excludedFileList> -f <frequency> -h <homozygous> -r <pearsonCorrelation> -p <primaryCNVR>
-m <mergedCNVR>
required options:
-l|--normalizedFileList  individual normalized copy number files list, with absolute path
-e|--excludedFileList    the samples in this list will be excluede from CNVR detection,
                         their genotyping are reported based on the CNVR boundaries defined by other samples.
                         This option is applicable to the outgroup individual
-f|--frequency           minimum frequency of gain/loss individuals when define a candidate CNV window
                         [recommend 0.1]
-h|--homozygous          minimum number of homozygous gain/loss individuals when define a candidate CNV window
                         [recommend 3]
-r|--pearsonCorrelation  minimum of pearson correlation coefficient between the two adjacent non-overlapping windows
                         during CNVR discovery
                         [recommend:]
                         0.5 for sample size (0, 30]
                         0.4 for sample size (30, 50]
                         0.3 for sample size (50, 100]
                         0.2 for sample size (100, 200]
                         0.15 for sample size (200, 500]
                         0.1 for sample size (500,+∞)
-p|--primaryCNVR         primary CNVR result
-m|--mergedCNVR          merged CNVR result
```

作用：通过综合考虑绝对拷贝数的分布、变异的频率及相邻窗口的显著相关性来初步确定 CNVR 的边界（primaryCNVR）。最后，将相邻且拷贝数在群体中分布显著相关的 CNVR 进一步合并得到最终的拷贝数变异检测结果（mergedCNVR）。

输入：上一步输出的文件名列表（一行一个样本，需包含绝对路径），默认位于 RD_normalized 下。

运行例子：对 list 文件记录的 10 个个体进行 CNVR 检测。其中 list 文件内容如下，exclude_list 为空文件。

```
/RD_normalized/ERR340328_mean_70.81_SD_10.84_sex_1
/RD_normalized/ERR340329_mean_62.00_SD_10.52_sex_1
/RD_normalized/ERR340330_mean_135.66_SD_13.96_sex_1
/RD_normalized/ERR340331_mean_128.76_SD_15.27_sex_1
/RD_normalized/ERR340332_mean_69.30_SD_10.19_sex_1
/RD_normalized/ERR340333_mean_132.30_SD_14.59_sex_2
/RD_normalized/ERR340334_mean_73.50_SD_10.16_sex_1
/RD_normalized/ERR340335_mean_72.52_SD_10.03_sex_2
/RD_normalized/ERR340336_mean_124.12_SD_13.24_sex_1
/RD_normalized/ERR340337_mean_131.00_SD_14.74_sex_1
```

```
bash CNV.Discovery.sh -l list -e exclude_list -f 0.1 -h 1 -r 0.5 -p primaryCNVR -m mergeCNVR
```

输出：上述运行完成后，会在当前目录下产生两个文件名为 primaryCNVR，mergeCNVR 的文本文件。

参数详解：

-l 个体经绝对拷贝数校正后的结果文件列表（绝对拷贝数校正后的结果默认位于工作目录下的 RD_normalized 目录，列表中需要包含绝对路径）。注：最终结果文件样本顺序与该列表样本顺序一致，故可以将样本按照群体进行排序后再进行后续计算。

-e 位于该列表中的样本将不被用于 CNVR 的检测，但结果文件会记录这些样本的绝对拷贝数。这个选项适用于外群或者低质量的珍贵样本（文件格式如-l）。注：一个空文件代表所有个体将被用于 CNVR 的检测。

-f 当一个窗口有超过该频率的个体绝对拷贝数与正常拷贝（“1”）显著差异（杂合删除或者杂合复制）时，就定义该窗口为候选拷贝数变异窗口。

-h 当一个窗口有超过该频数的个体绝对拷贝数与正常拷贝（“1”）显著差异（纯合删除或者纯合复制）时，就定义该窗口为候选拷贝数变异窗口。

注：在定义候选拷贝数变异窗口时只需要满足-f 与-h 中的任意一个即可。

-r 在定义 CNVR 时，如果相邻（没有 overlap）候选拷贝数变异窗口的绝对拷贝数的相关系数高于该值将被合并。

注：推荐使用显著水平为 0.01 的 pearson 相关系数。该值越高拷贝数检测的准确性越高，但获得的拷贝数变异区域越碎片化。

基因型判定

```
/home/CYD/software/anaconda3/bin/python /home/CYD/script/CNVcaller/CNVcaller-1.0.0/GMM_Genotype.py --help
Usage: GMM_Genotype.py [OPTIONS]

  Clustering the input samples into genotypes uses Gaussian mixture modes.

Options:
  --cnvfile TEXT        input cnvr file which generated from CNV.Discovery.sh
  --outprefix TEXT      output prefix of genotyped file
  --merge / --no-merge  merge all duplication types to one type, default is
                        false
  --nproc INTEGER       number of process will be used
  --help                Show this message and exit.
```

作用：利用混合高斯模型将每个样本的拷贝数归入不同的基因型分类，并以 VCF 格式输出，以方便后续通过全基因组关联分析挖掘与重要经济性状有关的拷贝数变异。

输入：上一步的输出结果（mergeCNVR），包含 CNVR 的区域所有样本的绝对拷贝数。

例子：使用 24 个进程对上一步的输出结果（mergeCNVR）进行基因分型并生成 VCF 文件。

python Genotype.py --cnvfile mergeCNVR --outprefix genotypeCNVR --nproc 24

输出：genotypeCNVR.vcf 和 genotypeCNVR.tsv。二者均包含每个个体的基因分型结果，以及每个 CNVR 基因分型的似然值和轮廓系数，区别在于 genotypeCNVR.vcf 为 VCF 格式，genotypeCNVR.tsv 为 tab 分割的表格格式。此外，genotypeCNVR.tsv 还包含更多的字段用于存放分类结果的统计信息，如每一类的频数及对应的绝对拷贝数的平均值与标准差。

VCF 各字段含义如下：

CHROM:CNVR 所在的序列名称。

POS：CNVR 的起始坐标位置。

ID：CNVR 的编号，格式为：序列标号:起始位置-终止位置。

ALT：变异类型，包括 CN0、CN1、CN2 和 CNH，分别代表零个拷贝、一个拷贝、两个拷贝和超过两个拷贝。

INFO：包含 CNVR 的终止位置（END），CNVR 的变异类型（SVTYPE），基因分型的对数似然值（LOGLIKELIHOOD）和轮廓系数（SILHOUETTESCORE）。

FORMAT：每个个体基因分型结果的输出格式，GT 和 CP 分别代表个体的基因分型结果和绝对拷贝数。

参数详解：

--cnvfile CNVR 结果文件，包含全部样本的拷贝数信息，由上一步的 CNV.Discovery.sh 得到。

--outprefix 输出结果文件的前缀，默认会输出两个文件，其中，后缀为 tsv 的文件记录了分类结果的基本统计信息，方便后续过滤低质量的 CNVR；后缀为 vcf 的文件为常规 VCF 格式，各字段具体含义见 VCF 注释部分。

可选参数：

--merge 为了得到更多的双等位 CNVR 用于后续分析，可以使用--merge 选项。使用后，会增加一个后缀为 _merge.vcf 的输出文件，类似<outprefix>.vcf 文件，区别在于<outprefix>_merge.vcf 中把所有重复算作一种变异类型

--nproc 程序使用的进程数，默认为单进程，使用此参数可显著减少程序运行时间，但会增加内存消耗

怎么定义一个窗口是否为拷贝数变异窗口？

在定义一个窗口是否为拷贝数变异窗口时，CNVcaller 首先需要对各个样本设定阈值。考虑

到测序过程中，可能会有一定比例的样本由于各种系统偏差导致测序 reads 出现系统偏斜，从而造成窗口读段数不是随机分布，即窗口读段数的标准差会明显升高。CNVcaller 默认群体中至少 75%的样本是测序质量较高的样本。根据经验，校正后滑动窗口 reads 数的变异系数（coefficient of variation，CV）一般要低于 0.2。对于测序质量较高（或者 CV 值较低）的前 75%的样本，CNVcaller 默认拷贝数缺失与重复的阈值分别设为：1-0.35 和 1+0.35；针对测序质量相对差（或者 CV 值较大）的后 25%的样本，CNVcaller 会通过样本的 CV 与所有样本 CV 的四分之三分位数的比值计算得到该样本的矫正系数 γ，并将拷贝数缺失与重复的阈值分别设为：1-0.35* γ 和 1+0.35* γ 。**对于不符合哈迪温伯格平衡的群体（如大豆、小麦杂合变异比例极低的纯系群体），CNVcaller 默认拷贝数缺失与重复的阈值分别为：1-0.75 和 1+0.75**。

关于频率，CNVcaller 通过分别统计基因组每个窗口杂合变异和纯合变异的样本数，默认杂合变异或纯合变异样本占所有样本 10%以上，即 CNV 在二倍体群体中的变异频率超过 5%，该窗口就被判定为拷贝数变异窗口。**对于纯系群体，CNVcaller 默认纯合变异个体数超过 2 的窗口为拷贝数变异窗口**。

怎么过滤低质量的拷贝数变异？

在拿到最初的拷贝数变异集合后，用户可能还需要经过一系列的筛选才能得到最可信的集合，这里和大家分享一些过滤标准；

1. 当某样本 CV 超过 50%时，可舍弃该样本。
2. 通过 CNVR 的长度进行筛选，一般情况下，长度越长准确性越高。
3. 根据基因分型后的结果进行过滤（可以参考轮廓系数，轮廓系数越大，说明分类效果越好），分类模式越明显，准确性越高。

如何选定合适的窗口大小？

窗口大小对于 CNVcaller 是一个比较重要的参数。窗口设置过大，对于短的 CNV 检测效率会低；窗口设置过小，结果的假阳性率会升高。建议挑选 2~3 个个体在不同长度窗口大小下分别计算标准化后的 CV，保证 CV≤0.3 即可。

如何用于纯系群体？

对于大豆，小麦这样的杂合变异比例极低的群体，CNVcaller 根据纯合变异个体数定义拷贝数变异窗口。因此，在运行 CNV.Discovery.sh 时，-f 参数需设为 2，-h 参数（纯合变异个体数）由用户自己设置，默认是 3。