Canu Quick Start

canu包含3阶段, correction阶段将提高read中碱基准确性; trimming阶段将会过滤read为高质量的序列(trim reads to the portion that appears to be high-quality sequence), 去除质疑区域, 例如SMRTbell adapter; 组装阶段将read排列成contig, 生成一致性序列(consensus sequence)同时构建alternate paths graph.

对于真核基因组, 超过20x的覆盖度足以超过当前hybrid methods, 同时推荐至少满足30x到60x覆盖度. 覆盖度越多, 组装效果也好.

canu能够恢复不完整的组装, 可以恢复由于系统内存不够或其他原因终止的组装. canu通过检查组装路径中的文件判断接下来要进行操作. 建议重启时不要修改canu参数.

unitigs (High quality contigs formed from unambiguous, unique overlaps of reads)

Assembling Pacbio or Nanopore data

Pacific Biosciences(SMRTpipe software software to convert the raw format to fastq) Esherichia coli K12, 223MB

`curl -L -o pacbio.fastq http://gembox.cbcb.umd.edu/mhap/raw/ecoli_p6_25x.filtered.fastq`

Nanopore Esherichia coli K12, 243MB

`curl -L -o oxford.fasta http://nanopore.s3.climb.ac.uk/MAP006-PCR-1_2D_pass.fasta`

默认条件下, canu将修正测序错误, 然后修剪reads, 最后组装成unitigs. canu需要知道大概基因组的大小(从而决定输入reads的覆盖度)和源自何种测序仪

PacBio:

```
canu \
 -p ecoli -d ecoli-pacbio \
 genomeSize=4.8m \
 -pacbio-raw pacbio.fastq
```

Nanopore:

```
canu \
 -p ecoli -d ecoli-oxford \
 genomeSize=4.8m \
 -nanopore-raw oxford.fasta
```

输出文件和中间文件将会保存在'ecoli-pacbio'和'eccoli-nanopore'路径中. 同时中间文件将会写入'correction', 'trimming'和'unitigging'中. 输出文件命名为'-p'前缀, 例如'ecoli.contigs.fasta'和'ecoli.contigs.gfa'

Assembling With Multiple Technologies and Multiple Files

canu可使用来自多个输入文件的reads, 其格式也可以混合存在, 例如组装来自10x pacbio和10x nanopore的fastq数据

```
curl -L -o mix.tar.gz http://gembox.cbcb.umd.edu/mhap/raw/ecoliP6Oxford.tar.gz
tar xvzf mix.tar.gz

canu \
 -p ecoli -d ecoli-mix \
 genomeSize=4.8m \
 -pacbio-raw pacbio.part?.fastq.gz \
 -nanopore-raw oxford.fasta.gz
```

Correct, Trim and Assemble, Manually

有时分布进行更有意义. 这将在同一套校正和过滤后的reads上尝试多种unitig construction参数, 或直接跳过过滤而直接组装.

首先校正pacbio raw reads:

```
canu -correct \
  -p ecoli -d ecoli \
  genomeSize=4.8m \
  -pacbio-raw  pacbio.fastq
```

然后过滤校正后reads:

```
canu -trim \
  -p ecoli -d ecoli \
  genomeSize=4.8m \
  -pacbio-corrected ecoli/ecoli.correctedReads.fasta.gz
```

最后组装过滤后的输出, 两次采用不同严格的重叠参数(correctedErrorRate):

```
canu -assemble \
  -p ecoli -d ecoli-erate-0.039 \
  genomeSize=4.8m \
  correctedErrorRate=0.039 \
  -pacbio-corrected ecoli/ecoli.trimmedReads.fasta.gz

canu -assemble \
  -p ecoli -d ecoli-erate-0.075 \
  genomeSize=4.8m \
  correctedErrorRate=0.075 \
  -pacbio-corrected ecoli/ecoli.trimmedReads.fasta.gz
```

这里采用了不同的输出路径`-d`, 不能够在相同的工作路径进行多个canu运行.

Assembing Low Coverage Datasets

canu可以组装最低20x覆盖度的测序数据, 这里组装20x 的S.cerevisae(215M). 在组装过程中调整correctedErrorRate来配合较低的质量校正reads:

```
curl -L -o yeast.20x.fastq.gz http://gembox.cbcb.umd.edu/mhap/raw/yeast_filtered.20x.fastq.gz

canu \
 -p asm -d yeast \
 genomeSize=12.1m \
 correctedErrorRate=0.105 \
 -pacbio-raw yeast.20x.fastq.gz
```

Trio Binning Assembly

Can has support for using parental short-read sequencing to classify and bin the F1 reads. This example demonstrates the functional using a synthetic mix of two Esherichia coli datasets. First download the data:

```
curl -L -o K12.parental.fasta https://gembox.cbcb.umd.edu/triobinning/example/k12.12.fasta
curl -L -o O157.parental.fasta https://gembox.cbcb.umd.edu/triobinning/example/o157.12.fasta
curl -L -o F1.fasta https://gembox.cbcb.umd.edu/triobinning/example/pacbio.fasta

trioCanu \
 -p asm -d ecoliTrio \
 genomeSize=5m \
 -haplotypeK12 K12.parental.fasta \
 -haplotypeO157 O157.parental.fasta \
 -pacbio-raw F1.fasta
```

The run will produce two assemblies, ecoliTrio/haplotypeK12/asm.contigs.fasta and ecoliTrio/haplotypeO157/asm.contigs.fasta. As comparison, you can try co-assembling the datasets instead:

```
canu \
 -p asm -d ecoliHap \
 genomeSize=5m \
 corOutCoverage=200 "batOptions=-dg 3 -db 3 -dr 1 -ca 500 -cp 50" \
-pacbio-raw F1.fasta
```

and compare the contiguity/accurary. The current version of trioCanu is not yet optimized for memory use so requires adjusted parameters for large genomes. Adding the options:

```
gridOptionsExecutive="--mem=250g" gridOptionsMeryl='--partition=largemem --mem=1000g'
```

should be sufficient for a mammalian genome.

Consensus Accurary

针对PacBio数据，Canu consensus sequences 一般可以达到超过99%的一致性, 针对Nanopore, 其准确性依赖于pore和basecaller版本, 但是一般也可实现98%的一致性.准确性的提供可通过polishing the contigs with tools developed specifically for that task. 针对PacBio数据, 推荐使用Quiver；针对Oxford Nanopore数据, 使用Nanopolish . 如果同时拥有illumina reads时, Pilon 可用于提升PacBio或Oxford Nanopore组装.

What parameters can I tweak?

针对所有分析阶段:

`rawErrorRate` 为align two uncorrected reads时期待最大的差异.

`correctedErrorRate` 为align two corrected reads时期待最大的差异(等同于errorRate multiply 3).

`minReadLength` 和 `minOverlapLength`, 默认为舍弃所有短于1000bp的reads, 同时不考虑重叠短于500bp的情况. 增加`minReadLength`长度将提高缩短运行时间, 提高`minOverlapLength`将通过舍弃假重叠来会提高组装质量. 然而，增加过多将会迅速减少组装长度.

针对校正阶段:

`corOutCoverage` 控制校正后reads覆盖度的产出. 默认为40X, 但是针对不同情况, 实际产出为30X至35X的reads产出(根据测序数据量和基因组大小决定).

`corMinCoverage`, 宽松地控制校正reads的质量. 校正的reads作为其他reads的一致性reads而产出; 该值为一致性序列产出的最小覆盖度. 默认值根据输入read的覆盖度决定: 低于30X的输入为0X,高于30X的为4X.

差别位点:

```
read1:   ATGACGTGATCGTAGCTGATCGTCGTTGGGAA
read2:       CGTGATCGTAGCTGATCGTCG TGGGAAACAG
read3:           ATCGTAGCTGATCGTCG TGGGAAACAGATGA
read4:               TAGCTGATCGTCG TGGGAAACAGATGAATG
                                   ^
                                   |
                                差别位点
```

数据校正:

```
read1:   ATGACGTGATCGTAGCTGATCGTCGTGGGAA
read2:       CGTGATCGTAGCTGATCGTCGTGGGAAACAG
read3:           ATCGTAGCTGATCGTCGTGGGAAACAGATGA
read4:               TAGCTGATCGTCGTGGGAAACAGATGAATG
```

数据过滤:

```
read1:   ATGACGTGATCGTAGCTGATCGTCGTGGGAA
read2:       CGTGATCGTAGCTGATCGTCGTGGGAAACAG
read3:           ATCGTAGCTGATCGTCGTGGGAAACAGATGA
read4:               TAGCTGATCGTCGTGGGAAACAGATGAATG
         |_________|
           低质量区
```

去除低质量:

```
read1:   GTAGCTGATCGTCGTGGGAA
read2:   GTAGCTGATCGTCGTGGGAAACAG
read3:   GTAGCTGATCGTCGTGGGAAACAGATGA
read4:    TAGCTGATCGTCGTGGGAAACAGATGAATG
```

组装:

```
---------------
        ----------------
                  --------------
    |          |           |
    v          v           v
-----------------------------------
```

针对组装阶段:

`utgOvlErrorRate` 为优化速度的必要选项. 忽略高于该错误率的重叠情况. 设置过高将浪费计算时间, 设置过低将降低组装质量会舍弃低质量reads见真正的重叠.

`utgGraphDeviation`和`utgRepeatDeviation`, 分别为用于构建contig的重叠质量值或者在假重复joins处断裂contigs的质量值. 两者均为最长重叠的平均错误率的deviation.

`utgRepeatConfusedBP` 控制真正重叠(同一contig中两reads)和假重叠(不同contigs中的两reads)的相似性, 用于contig分开(need to be before the contig is split). When this occurs, it isn't clear which overlap is 'true' - the longer one or the slightly shorter one - and the contig is split to avoid misassemblies.

针对多倍体基因组:

略 **canu Documentation Release 1.8**

针对metagenomes:

其根本思想是使用所有的数据组装, 而不是默认使用最长的reads:

```
corOutCoverage=10000 corMhapSensitivity=high
corMinCoverage=0 redMemory=32 oeaMemory=32 batMemory=200
```

针对低覆盖度:

针对低于30X的覆盖度, 增加重叠时所允许的差异(针对PacBio, `correctedErrorRate=0.105`, 差异百分比可设置从4.5%到8.5%, 或更多; 针对Nanopore, `correctedErrorRate=0.16`, 差异百分比可设置为14.4%到16%, 或更多), 来配合较低的read校正. Canu将会自动减少`corMinCoverage`为0来校正尽可能多的reads.

针对高覆盖度:

针对高于60X的覆盖赋, 针对PacBio, `correctedErrorRate=0.040`, 重叠差异可以设置4.5%到4.0%; 针对Nanopore, `correctedErrorRate=0.12`, 重叠差异可以设置为14.4%到12%. 这样就保证了仅使用较好的校正后reads, 这将提高运行速度, 同时不会改变组装连续性.

My asm.contigs.fasta is empty, why?

Canu输出3个组装了的输出序列: .conitgs.fasta, .unitigs.fasta, .unassembled.fasta. contigs文件为主要输出, unitigs文件为主要分割为不同路径的输出, unassembled文件为剩下的部分.

参数`contigFilter`控制最低的初始contigs覆盖度. 默认, 初始contig超过50%的长度具有低于3X覆盖度将会定义为'unassembled', 同时从组装中去除, `contigFilter="2.0 1.0 0.5 3"`. 可以将最后的数字从3改为0来取消过滤(这意味, 如果超过50%的长度低于0X覆盖度将舍弃)

Why do I get less corrected read data than I asked for?

由于校正阶段, 一些嵌合reads由于不充分证据来生成校正后reads而被过滤掉. 一般而言, 会带来25%的reads丢失. 设置`corMinCoverage=0`将会产出所有reads, 即使低质量的reads. Canu将会在组装阶段前的过滤阶段过滤掉这些reads.

What circular element is duplicated/has overlap?

任何环状单元都可以发生. 可基于Canu如何构建contigs而带来至多reads长度的重叠. Canu provides an alignment string in the GFA output which can be converted to an alignment to identify the trimming points.

或使用MUMmer, 自身比对:

```
nucmer -maxmatch -nosimplify tig00000099.fa tig00000099.fa
show-coords -lrcTH out.delta
```

to find the end overlaps in the tig. The output would be something like:

```
1 1895    48502   50400   1895    1899    99.37   50400   50400   3.76    3.
↪77    tig00000001     tig00000001
48502     50400   1       1895    1899    1895    99.37   50400   50400   3.
↪77    3.76    tig00000001     tig00000001
```

means trim to 1 to 48502. There is also an alternate writeup.

/ˈreɪt ʌp/
n. 报导,评论(尤指捧场文章)
账面价值的提高;资产的过高估价

My genome is AT(or GC)rich, do I need to adjust parameters? What about highly repetitive genomes?

针对细菌基因组, 无需采用任何参数(一般而言).

针对具有显著AT/GC比率倾斜的重复性基因组, the Jaccard estimate used by MHAP is biased. 设置`corMaxEvidenceErate=0.15`足以校正偏差.

一般而言, 针对高覆盖度的重复基因组(例如植物), 可通过设置以上参数获益, 将会舍弃重复匹配, 加速组装, 有时还会改善unitigs.

Canu

Canu用于组装来自PacBio RS II或Oxford Nanopore MinION的测序reads，该软件很多设计和代码来自celera-assembler

canu组装主要包括三个阶段(correction, trimming, uniting construction)，每一阶段都包含许多步骤。

```
canu [-correct | -trim | -assemble | -trim-assemble] \
  [-s <assembly-specifications-file>] \
   -p <assembly-prefix> \
   -d <assembly-directory> \
   genomeSize=<number>[g|m|k] \
   [other-options] \
   [-pacbio-raw | -pacbio-corrected | -nanopore-raw | -nanopore-corrected] *fastq
```

`-p`设置中间和输出文件名称前缀，强制性选项. 若`-d`选项没有提供，则在当前目录运行, 否则将创建组装目录，并在该目录运行.

`-s`选项用于指定包含一系列文件的参数('spec'). 这些参数将优先与所有命令行参数, 用于提供通用组装参数.

默认条件下，所有三个阶段分析都会执行, 同时也可以仅运行其中一个进程`-correct`, `-trim`, `-assemble`. 可使用这些选项先修正所有reads，然后尝试不同的组装过程. 同时提供`-pacbio-corrected`和`-nanopore-corrected`选项用于仅进行`-trim`和`-assemble`分析过程.

[Parameters][https://canu.readthedocs.io/en/latest/parameter-reference.html#parameter-reference](参数)格式为key=value对形式配制组装(assembler). 用于设置运行时间参数(e.g. memory, threads, grid), 算法参数(e.g. error rates, trimming aggressiveness), 和是否进行全部阶段分析(e.g. don't correct errors, don't search for subreads). 其中一个参数是必须的, genomeSize(单位为baes, with common SI prefixes allowed, for example, 4.7m 或 2.8g; see genomeSize)

Reads通过选型告知reads的生成信息提供给canu, 例如`-pacbio-raw`表明reads由PacBio RS II设备测序, 同时没有进行加工处理. 每一个提供的reads文件都将看成reads的'library'. 这些reads应相同的步骤时间生成(物理上而言), 但可以是不同的测序批次. 每一个library都拥有一套paramers设置, 例如, trim的程度等. 为精确设置library参数, 通过设置文本'gkp'文件描述library和输入文件实现.

Read-files包含序列数据可以是fastq或fasta格式(或者两者同时). 文件可以是未压缩的, gzip, bzip2或xz压缩格式.

Canu, the pipeline

canu pipeline为实际计算过程. 所有三个阶段都遵循相同模式(read correction, read trimming和unitig construction)

- Load reads into the read database, gkpStore.
- Compute k-mer counts in preparation for the overlap computation.
- Compute overlaps.
- Load overlaps into the overlap database, ovlStore.
- Do something interesting with the reads and overlaps.
  - The read correction task will replace the original noisy read sequences with consensus sequences computed from overlapping reads.
  - The read trimming task will use overlapping reads to decide what regions of each read are high-quality sequence, and what regions should be trimmed. After trimming, the single largest high-quality chunk of sequence is retained.
  - The unitig construction task finds sets of overlaps that are consistent, and uses those to place reads into a multialignment layout. The layout is then used to generate a consensus sequence for the unitig.

Module Tags

Execution Configuration

Error Rates

Fraction Error	Percent Error
0.01	1%
0.02	2%
0.03	3%
.	.
.	.
0.12	12%
.	.
.	.

Canu错误率指的是比对的两个reads的差异百分率, 而不是单个read的错误率, 也不是reads的变异量. 这些错误率使用两个不同的方式: 用于限制重叠(overlaps)的产生, 例如, 不去计算超过5%差异的重叠; 同时告知算法使用怎么样的重叠(overlalps).

总共有7个错误率, 3个错误率控制overlap产生(vorOvlErrorRate, obtOvlErrorRate和utgOvlErrorRate), 4个错误率控制算法(corErrorRate, obtErrorRate, utgErrorRate, cnsErrorRate)

一般而言, 两个meta选项设置error rates用于未修正的reads(rawErrorRate)或修正后的reads(correctedErrorRate).

Parameter	PacBio	Nanopore
rawErrorRate	0.300	0.500
correctedErrorRate	0.045	0.144

实际上, 只有correctedErrorRate常被修改, 见[Canu FAQ specific suggestions][https://canu.readthedocs.io/en/latest/faq.html#tweak]

Minimum Lengths

`minReadLength` 当进行assembler和trimming reads时舍弃短于该值当reads

`minOverlapLength`重叠(overlap)短于该值取消

Overlap configuration

装配过程中计算量最大也是最复杂的配置. 在module tag中共有8个不同的overlapper配置. 针对ovl和mhap，拥有一个整体的配置，和三个指定的配置用于三个阶段.

和'grid configuration'一样, overlap configuration使用'tag'前缀指明每一个选项. 该离子中的标签为'cor', 'obt', 'utg'

- To change the k-mer size for all instances of the ovl overlapper, ‘merSize=23’ would be used.
- To change the k-mer size for just the ovl overlapper used during correction, ‘corMerSize=16’ would be used.
- To change the mhap k-mer size for all instances, ‘mhapMerSize=18’ would be used.
- To change the mhap k-mer size just during correction, ‘corMhapMerSize=15’ would be used.
- To use minimap for overlap computation just during correction, ‘corOverlapper=minimap’ would be used. The minimap2 executable must be symlinked from the Canu binary folder (‘Linux-amd64/bin’ or ‘Darwin-amd64/bin’ depending on your system).

Ovl Overlapper Configuration

Ovl Overlapper Parameters

Mhap Overlapper Parameters

Minimap Overlapper Paramters

Outputs

canu运行中, 输出状态信息, 执行日志, 和一些分析. 大多以前缀`<prefix>.repor`.

[Canu Pipeline][https://canu.readthedocs.io/en/latest/pipeline.html]

Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation.

[Canu Parameter Reference] [https://canu.readthedocs.io/en/latest/parameter-reference.html]

[Canu Command Reference] [https://canu.readthedocs.io/en/latest/command-reference.html]

[Software Background] [https://canu.readthedocs.io/en/latest/history.html]

Miscellaneous

{prefix}MhapSensitivity <string="normal">

Coarse sensitivity level: ‘low’, ‘normal’ or ‘high’. Based on read coverage (which is impacted by genomeSize), ‘low’ sensitivity is used if coverage is more than 60; ‘normal’ is used if coverage is between 60 and 30, and ‘high’ is used for coverages less than 30.