从本期开始，我们将陆续推出Cloudera 工程博客的翻译，觉得看中文翻译别扭的童鞋，戳这里看原文咯：http://blog.cloudera.com

**基因组分析工具包：现在可以使用Apache Spark进行数据处理了**

2016年04月06日/作者：Tom White

**最新版本的基因组分析工具包（这是一款用于分析高通量DNA测序数据的开源代码框架）的用户，现在可以选择Apache Spark进行数据处理。**

自从人类基因组计划（Human Genome Project）于2000年发布了第一套人类基因组的序列草图，基因组测序成本已成指数下降，从过去的每个基因组大约1亿美元降低到今天的大约1千美元。在同一时期，我们也已经看到大数据技术（比如Apache Hadoop）的存储和处理能力获得了显著的增长。

因此，就非常适合使用Hadoop生态系统的工具来完成基因组学研究工作，这就是为什么Cloudera公司选择与布罗德研究所（[Broad Institute](https://www.broadinstitute.org/)）等业界合作伙伴进行合作的原因。我们很高兴地宣布，alpha版本的基因组分析工具包（GATK）version 4已经运行在Apache Spark上。

基因组学的数据处理，与任何行业的数据处理类似，通过数据管道的模式进行分析处理。一种用于DNA测序的数据管道如图1所示。

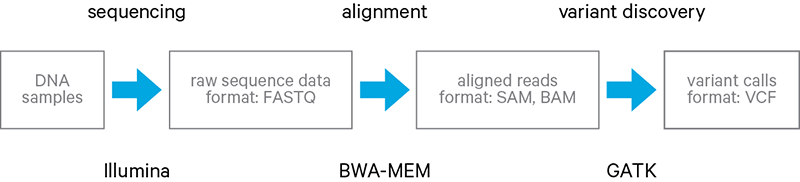


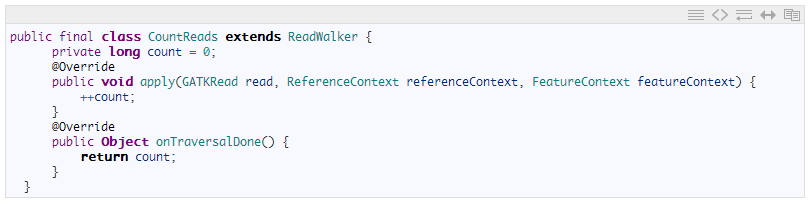
图1、DNA测序数据管道

该数据管道是从DNA样本入手的；这些DNA样本是由一台机器进行测序，然后输出了一个含有DNA序列片段的文件（由字母A、C、G和T组成）。此时，原始序列数据并不是非常有用，因为该序列片段在基因组中未能占有一席之地。因此，我们使用了一款称之为调整器（aligner）的软件来调整该片段在参考基因组的位置。（该参考基因组是人类基因组计划（Human Genome Project）的产物。）在这个阶段，对该个体而言DNA序列是已知的，但接下来的问题通常是：“请问该DNA序列与其他个体有何不同，”数据管道中的这一步意味着相对于参考基因组寻找序列中的变异：从概念上讲，您可以把它看作在序列和参考基因组上运行Unix diff命令。其输出是一组针对个体的变异调用。此时，图1中的数据管道停止运行，但是在实践中，变异调用数据是研究者开展后续分析研究的原材料。

该基因组分析工具包（GATK）涵盖了数据管道的变异发现环节。变异发现本身就是由许多步骤组成的，并且GATK提供了运行这些步骤的相关工具。（具体步骤请参见GATK最佳实践文档。）GATK版本4的新特性在于现在提供了这些工具的Spark实现，从而可以允许该数据管道的变异发现环节在Spark集群上大规模运行。如需了解其中的具体内容，请更详细参阅GATK。

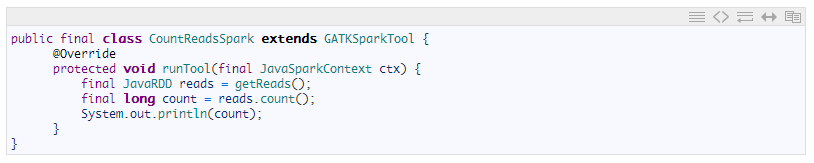
**GATK内部特性**

该GATK具有一种Walker的概念，实质上是一个回调迭代程序API。例如，ReadWalker在基因组中迭代所有测定的序列（Reads），并针对每一次测定的序列（Reads）调用回调接口的一个方法。（*测定的序列（Reads）（*read*）*是已与参考基因组调整对齐的DNA的短片段。）关于Walker API最吸引人的特性就是其可以让用户很容易地编写他们自己的工具。其中最简单的一种方案就是计数一个文件中测定的序列（Reads）的数目，如下所示：



apply（）方法会递增计数器的计数；然后，一旦到达文件的结尾，将调用TraversalDone（）函数来报告计数。CountReads是一个串行程序，所以只有一个进程用于计数一个文件中的测定的序列（Reads）数目（或甚至多个文件中的测定的序列（Reads）数目）。这种方法具有过程简单的优点，但是也存在速度缓慢的缺点，特别是在处理TB字节的大量数据时这一缺点非常显著。

Spark等效程序的结构有所不同，因为其不是一个回调函数，而是利用工具本身实施输入的处理。这里采用的是CountReads的分布式版本。（为简单起见，在GATK源代码中已经删除了一些样板文件代码。）



在这种情况下，测定的序列（Reads）通过Spark RDD（JavaRDD <GATKRead>）进行访问，然后调用Spark的内置函数count（）来触发一个Spark作业，旨在计数RDD中的条目数量。该输入将被分成多块（默认情况下，每块的大小为128MB），并且Spark 作业将并行运行每个分块的任务。对于利用Walker版本需要耗费数小时时间执行的作业，采用适度的Spark集群只需要数分钟时间就可以执行该作业。

测定的序列（Reads）的计数是非常繁琐的。GATK中的大多数工具是更为复杂的，并且无法非常容易地被移植到Spark中。现在让我们进行更深入地探讨：Mark Duplicates算法。

**Spark中的Mark Duplicates算法**

基因测序过程是一种固有的噪声过程，并且经常发生相同的DNA片段被测序多次的情况，从而导致出现多个测定的序列（Reads）重复的现象。因此，需要删除这些重复序列，使它们不会在数据管道存在过度的权重。Mark Duplicates算法可以用于发现和标记应被视为是相同的测定的序列（Reads）。

您怎么知道两个（或更多）测定的序列（Reads）是重复的？测定的序列（Reads）中包含几个构成了顺位信息的字段；如果这些字段对于一组测定的序列（Reads）是相同的，则他们将被视为是重复的。重复的测定的序列（Reads）是不相同的，因此，Mark Duplicates算法将根据该测定的序列（Read）的其他方面的信息（例如，质量测量结果）对一组重复序列中的每一个测定的序列（Read）进行评分。然后，在各组重复序列中，得分最高的测定的序列（Read）将保持不变，而其它测定的序列（Reads）将被标记为重复序列。

Mark Duplicates算法归结起来就是根据顺位信息字段对所有测定的序列（Reads）进行排序。使用Walker API来实现上述过程是非常困难的，因为所有测定的序列（Reads）不能全部装入内存中。该实现过程使用了一种自定义的存在磁盘上的排序算法，但即使是这样也只能利用一台机器。

由于Spark的出现，我们可以获得高度优化的分布式排序实现，这是一个优秀的Java API所蕴含的潜力。让我们来看看Mark Duplicates算法实现的核心部分。我们首先从按照测定的序列（Read）的组别和名称划分的测定的序列（Reads）入手（文件通常已经是这样排序的，但是如果没有这样排序，需要执行一个初始排序）。



接下来，我们将提取每个测定的序列（Read）的顺位信息字段以形成一个字符串，并且针对该值构造PairedEnds对象。测定的序列（Reads）通常是成对的，一对测定的序列（Reads）中的每一个都是从被测序的DNA片段的任一端开始的。一个PairedEnds对象对于一对测定的序列（Reads）只是一个封装而已。



然后，我们将执行分组操作，将具有相同顺位信息的所有测定的序列（Reads）（在PairedEnds对象中）集合在一起。



最后，对于每一组序列，可以进行评分和标记过程：



**通过Spark组合工具**

Mark Duplicates工具将把测定的序列（Reads）的最终RDD写入到一个输出文件中，以便由其他GATK工具执行进一步的后续处理。由于其输出是一个RDD，另一种选择是在一个单独的Spark作业中将所有的工具组合在一起，这样就不需要在文件系统中生成中间步骤文件。

对于最新的alpha版本GATK4，并非所有的工具都已经移植到Spark中，所以您还不能作为一个单独的Spark作业来运行整个数据管道。然而，这一状况在未来会有所改观，其目标是能够在Spark集群上同时运行调整器（例如，BWA-MEM）和GATK变异发现管道。

随着越来越多的项目开始采用Spark进行大规模的计算，在基因组学管道中集成工具的可能性正在与日俱增。[ADAM](http://bdgenomics.org/)是首次使用Spark作为基因组学研究平台的首个项目，并且该项目还使用Apache Parquet定义了基因组数据的文件格式。作为一个选项，GATK4已经可以读取和写入ADAM Parquet格式的数据。另外一种令人感兴趣的可能性是混合及匹配管道中的工具。例如，您可以使用ADAM或[Hammer Lab](https://github.com/hammerlab)（杰弗•哈梅巴赫（Jeff Hammerbacher）位于美国纽约西奈山医院（Mount Sinai Hospital）的实验室）的变异调用程序。

如果再看远一点，另一个目标是将数据管道产生的变异调用数据集加载进入Hadoop的本地存储引擎，例如Hive（Parquet格式）或Apache Kudu（正在孵化中）。这将使用户能够充分利用各种工具的优势进行分析，例如Apache Impala （正在孵化中）、Ibis、或Spark，甚至可以基于这些软件框架为科学家构建适宜的工具。

**参与开放基因码运动（Open Source Genomics）**

GATK4依然还很年轻，但其已经获得的成果却是非常令人鼓舞的。作为其潜力的一个实例，布罗德研究所（Broad Institute）有一支团队正在致力于研究*基因组拷贝数变异（*copy number variations*）*，这在基因组中是非常复杂的变异，并且其中一段DNA是重复的。即使是毫无Spark的现成经验，他们也能编写一个Spark工具（由于计算复杂性，从未在GATK3中尝试这一点），并且该Spark工具的运行速度也大大提升了，据估计相对于GATK3而言，其速度提升了一个或两个数量级。

针对新型Spark实现，业界已经有众多研究小组正致力于算法优化和运行时间调优：其中包括来自Cloudera、Cray、Google、IBM和Intel公司的开发人员。由于巨大的工作量，基因组学是非常好的针对尖端硬件和软件的实验平台，因此一些已经发现的软件错误就无足轻重了（例如，在Java 8上运行时，KRYO中的这个bug）。

Spark作为一种通用的分布式计算体系架构，可以支持云端和内部部署运行环境。我们Cloudera公司希望其他开发商将来能够参加像GATK在内的构建于Spark上的项目。如需了解更多关于GATK的信息，请访问<https://github.com/broadinstitute/gatk>上的GitHub资料库。

Tom White*是*Cloudera*公司的一名数据科学家，并且是Apache Hadoop项目管理委员会（PMC）的成员。*Tom White*是O'Reilly出版社出版的书籍*Hadoop: The Definitive Guide（Hadoop：权威指南）*的作者*。