1. 输入输出

输入：指定的进化树结构、指定的根节点基因组、参数文件。

1. 进化树结构：表示为Newick格式的字符串，除根节点之外的其他节点都带有分支长度(分支长度随机生成，0-1之间，且保留两位小数)；
2. 指定的根节点基因组：文件的第一行表示节点名，从第二行开始每一行表示一条染色体，染色体上的每两个基因之间用空格隔开，染色体末尾用“$”表示一条染色体结束。
3. 参数文件(共两行)：

第一行是一个整数sum，sum表示这棵树上所有的分支上发生的重排事件总次数。sum的取值遵循以下规则：

sum = (genes / 10 + 2 \* (leaves - 1) + chromosome);

其中，genes是根节点基因组中的基因数，leaves表示当前树中的叶子节点数，chromosome表示根节点基因组中的染色体条数。

每条分支上发生的事件总数，是根据分支长度之间的比例决定的，长度相对更大的分支也将被分配更多的重排事件。

第二行内容形如“a b c d”,表示每条分支上发生的四种重排事件(反转、转位、移位、分裂和合并)的比例。

输出：对于每个节点，生成它的基因组并给出生成它的过程记录。基因组文件名为“nodeName\_chromosomes.txt”，记录生成过程的文件名为“nodeName\_options.txt”。

二、源代码中主要文件功能说明

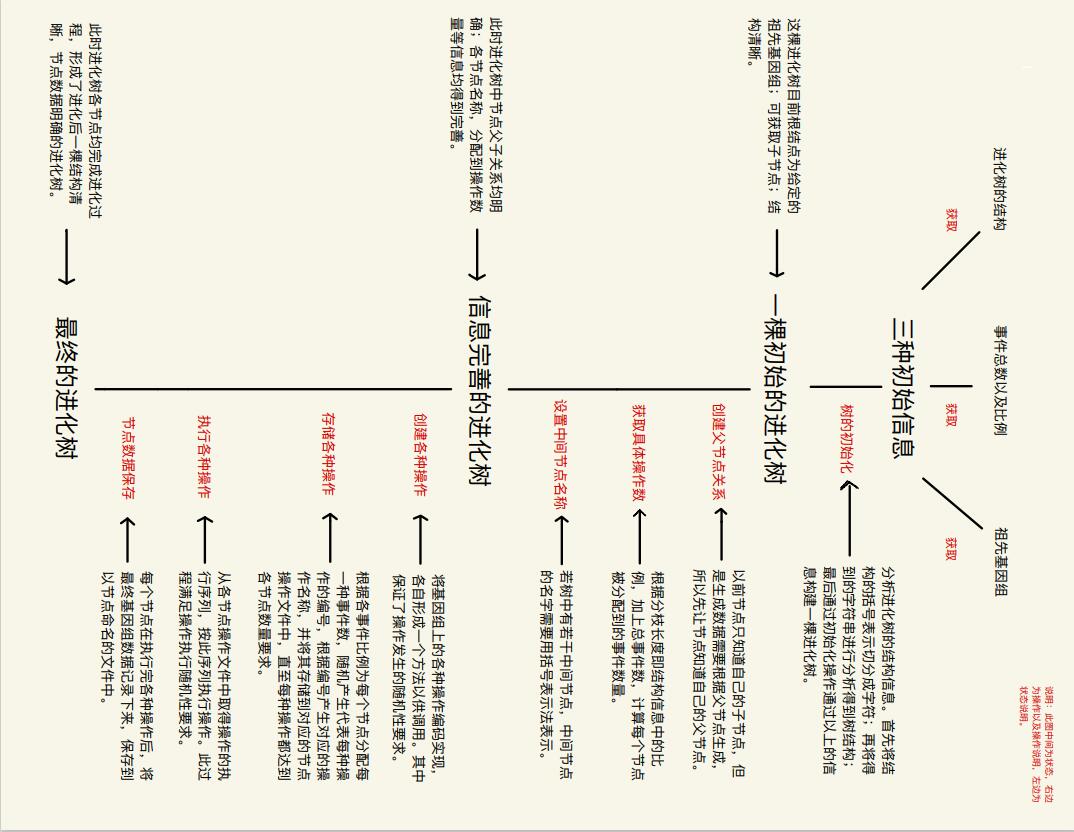
**Genome.java** ：基因组类，对应于对基因组进行各种处理的操作。其中包含，反转、移位、转位、分裂和合并四种主要操作。各个操作的具体实现均在代码中进行了标注。需要注意的是，分裂与合并操作是顺序进行的，进行一个分裂操作，接下来就要进行一个合并操作，这样做的目的是为了避免产生一个基因甚至没有基因的染色体。**Node.java** ：节点类，可对应于进化树中的每一个节点进行理解。这里包含对节点数据结构的定义，在包含获取、设置节点名字，获取、设置节点权值，获取、设置节点祖先，获取、设置节点名字，获取、设置节点孩子，设置、获取、分析染色体数据等众多基本功能以外，更为重要的是包含：根据输入数目比例随机创建各种操作、执行各种操作、打印各种操作，三种主要功能。

**Tree.java** : 进化树类，对应于对树的各种操作。为了解决从括号表示法到树结构的问题，引入了分析树结构函数analyzeStructure()；为了根据父节点生成子节点的基因组，引入了获取父节点函数konwfather()；当然，设置中间节点的名称、操作数，创建操作、创建基因组也都是通过这里的对树的遍历的基础上实现的。

以上为程序的主体类，下面是工具类说明：  
**Dataset.java** : 文件操作类。由于给定的祖先基因组与进化树的结构都是文件形式的，最终的结果（各个节点的基因组、操作）也都需要以文件的形式进行存储，这里专门写一个文件处理类对程序中涉及的主要文件操作统计进行处理。

**Util.java** : 工具包类。可能在各种操作的实现过程中，你会遇到各种各种函数，这些函数大多是在这里定义的，为了避免各种操作代码过于庞大，很多相似的操作都在这里定义成了一个个函数处理工具。

三、设计思路图示



**Tip:** 了解更多信息，可查看“/example\_for\_GSS/”路径下的内容，并阅读源代码。