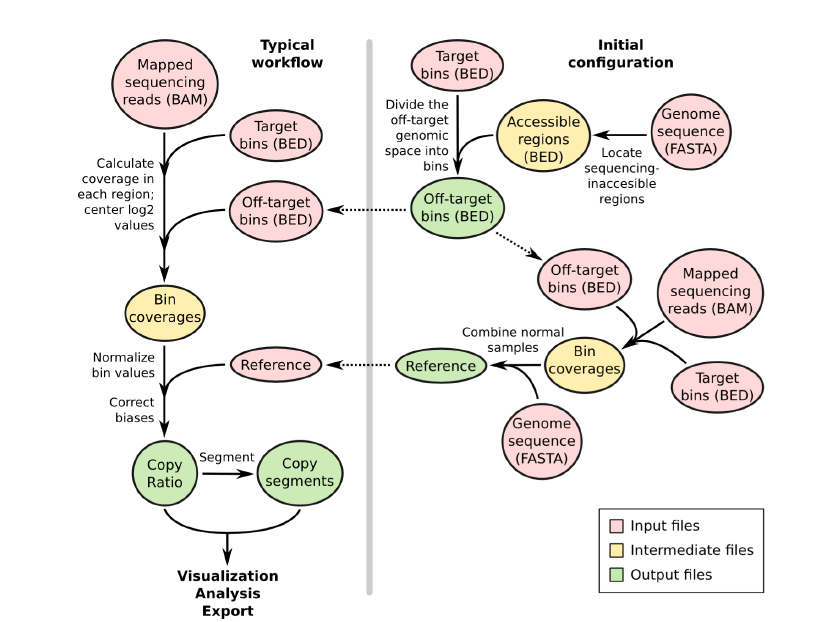
**CNVkit若干算法问题详解**

****

1. **target antitarget区域bin size确定**

target区域：

min size 20

max size 50000

antitarget区域：

min size 500

max size 1000000

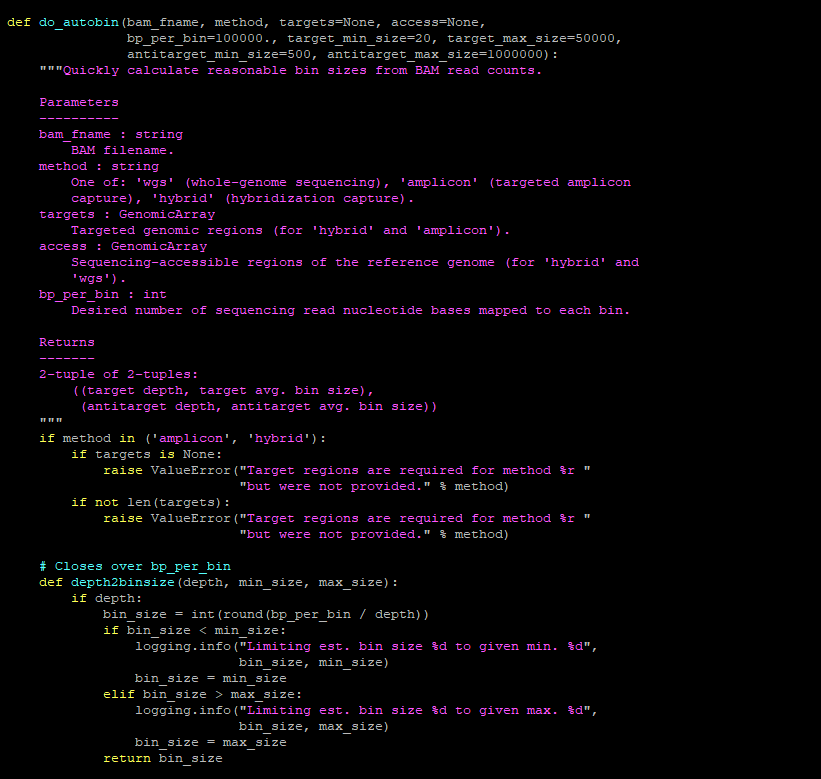
计算bin size公式：

bin\_size = int(round(bp\_per\_bin/depth)), 其中bp\_per\_bin初始化为100000

按以上计算公式，若

bin\_size < min\_size : bin\_size = min\_size

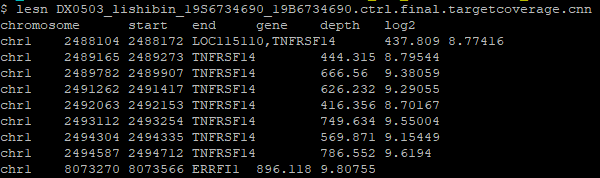
bin\_size > max\_size : bin\_size = max\_size

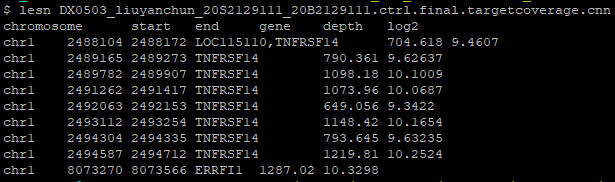


1. **多个control样本如何构建reference？**

该部分处理过程中会对每个样本的coverage深度（不做详细赘述，主要是通过计算每个bin区域碱基深度和/bin size长度）文件进行GC矫正和重复区域矫正（后面会详细介绍此部分），目前仅主要介绍如何通过多个control文件构建reference？weight权重如何计算？

假设我们得到的每个control样本的coverage文件信息，如下：

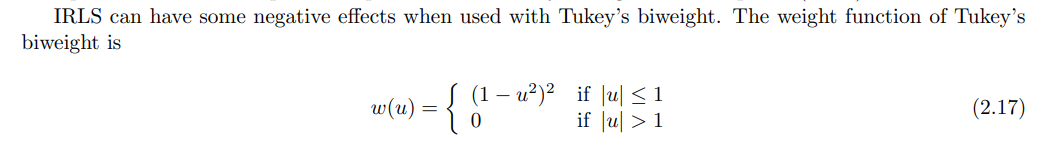




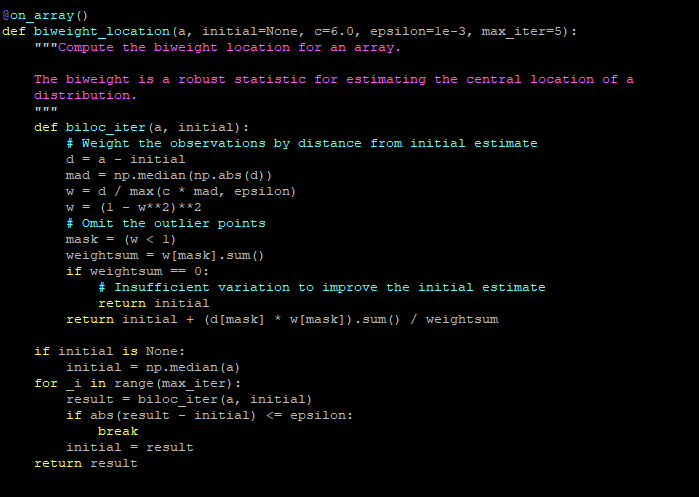
如何构建？

采用Tukey’s Biweight correlation, this calculates a robust average that is unaffected by outliers，也即是可以计算不受异常值影响的鲁棒性均值.

该方法的数学原理及公式推导可查到相关资料，如下仅列出权重weight相关公式：



用在此方法中是根据不同的control样本构建reference，排查异常bin深度造成的影响，如下：



计算过程如下：

* 计算过程举例：

5个对照中某bin的深度a=[120 130 154 118 9]，

initial（默认和初始均为中位数median）=120

实际bin深度与期望深度的差值d=[ 0 10 34 -2 -111]

d的绝对值的中位数mad=10

cutoff= c倍的mad（c默认为6）=60

权重w = (1 –(d/cutoff)2)2=[1.000 0.945 0.461 0.998 5.869]

* 离群值处理：w<1保留

可理解为仅将mad<d< mad且d≠0的变量计入取平均值的范围。

* 迭代次数

不超过5次

* 结果

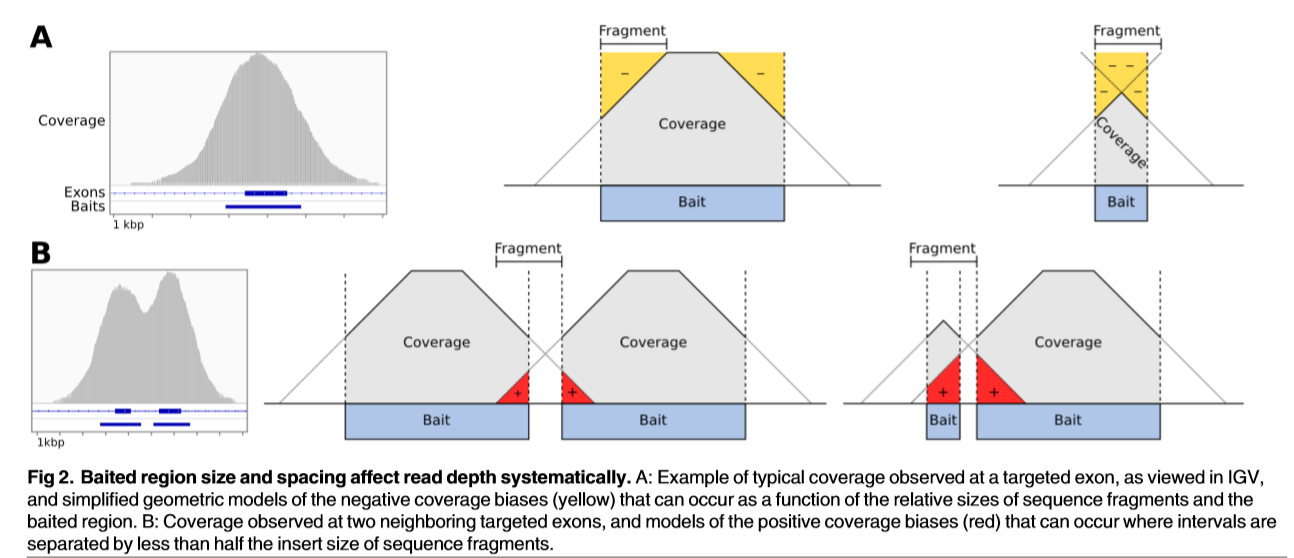
result = initial + (d[mask] \* w[mask]).sum() / weightsum

若 abs(result – initial) < = epsilon,返回result

经过上述计算，得出最终多个control pooling reference后的深度。

1. **bias correction矫正中涉及到三部分矫正**

* target density矫正(首先文献中如何计算的？)：

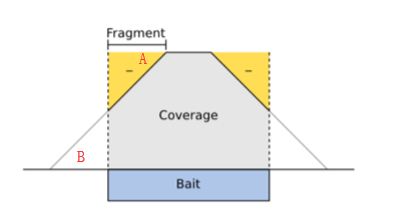


计算Shoulder effect：

每个bin区间的“shoulders”存在深度减少的现象，

A图中的黄色区域未测到，B图中的红色区域重复测到，需要减少其影响，主要采用的思想是近似看做三角形，计算面积占比

A：当t > i时，即如下图示：

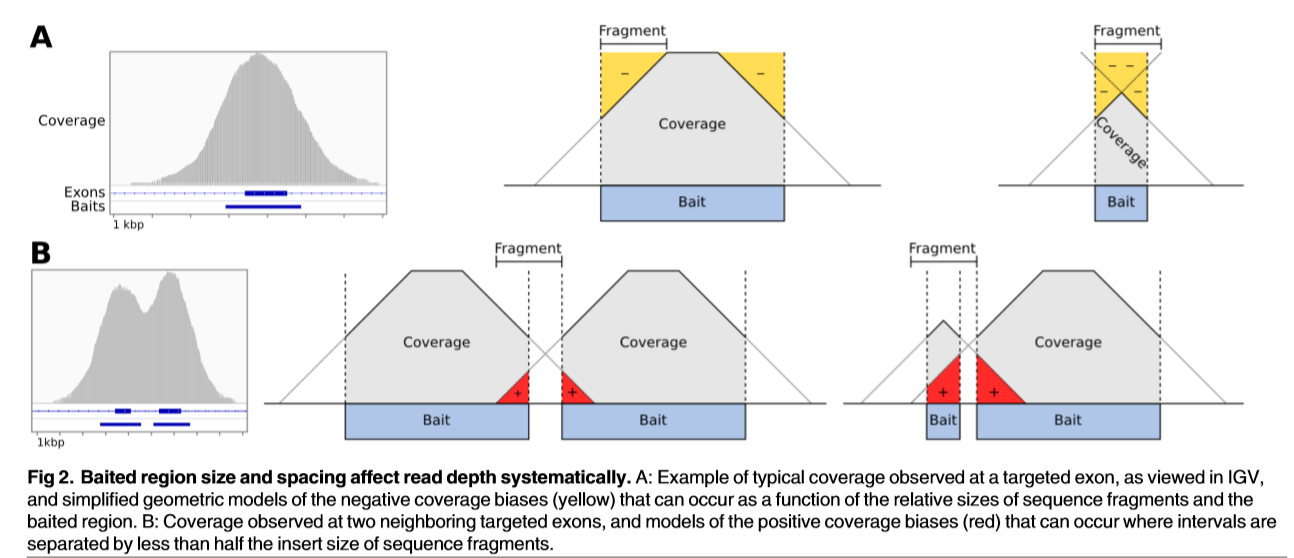


我们把整个深度分布的区域进行拉直化，即从圆滑曲线面积近似看做是某长方形面积，图中A和B是两个全等三角形，两个黄色区域的面积相等，

插入片段fragment为i，Bait（可以看做是bin）长度为t，coverage为高度h，所以，两个黄色区域面积占比为：

2\*（i\*h/2）/2 /(h\*t) = i/(2\*t)

当t < i时，即如下此图：



计算黄色区域面积占比，如下：

其中一部分计算公式和上面相同，主要是考虑减去圆圈框起来的两部分面积S，利用两个三角形相似，先计算S的两条直角边，其中一条为i-t，另一条利用三角形相似：

(i-t)/i = x/(h/2)

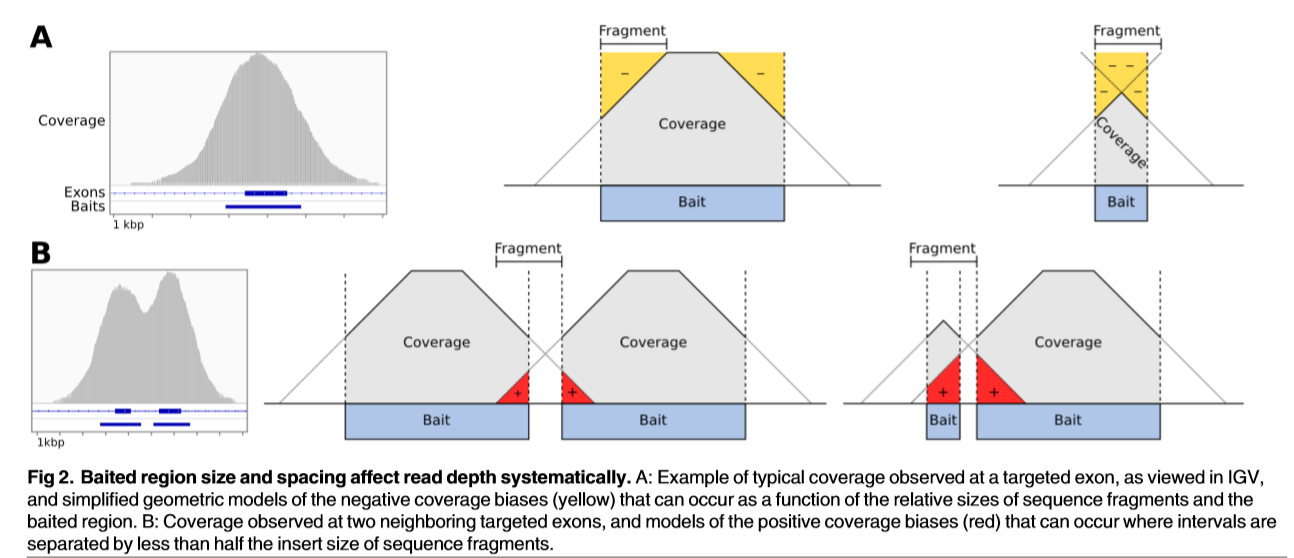
所以，x = h(i-t)/(2\*i)

所以其中一个小三角形的面积为：

h\*(i-t)^2/4\*i

然后我们可以计算得到此部分面积占比，最后可知在此种情况下，缺失部分的面积占比为：

计算完以上shoulder effect后，我们来计算flank effect，如下：



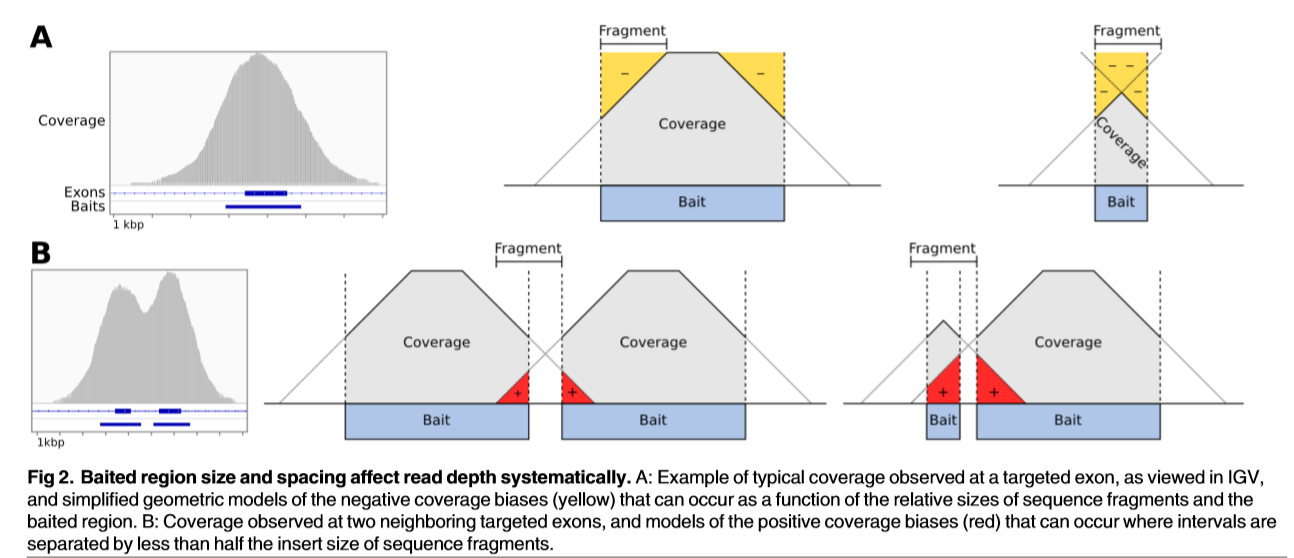
假设两个Bait之间的gap长度为g，Fragment长度为i，Bait长度为t，Coverage高度为h，

若g < i ,flank effect的面积为：

依然是利用三角形相似原理，先计算红色区域的其中一边，然后再计算红色区域的面积，最终计算占比，先看相似三角形：

计算出来x，然后计算红色三角形面积，最终计算占比为：

若t+g < i 时，



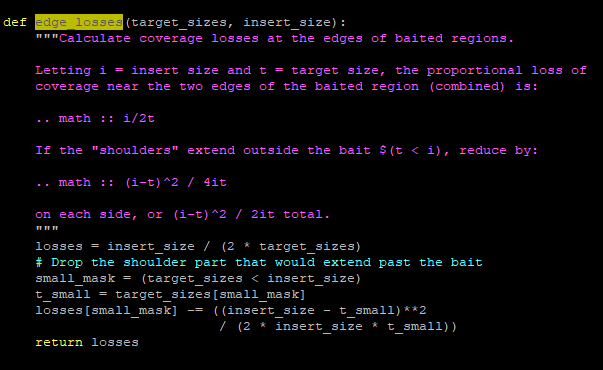
同之前shoulder effect计算，减去红色圈圈的面积，

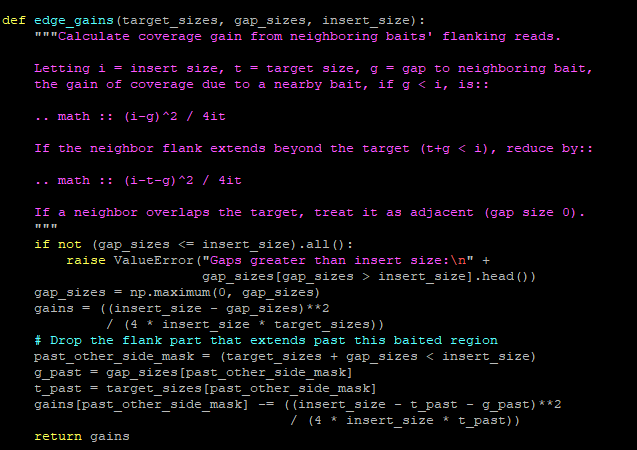
同样借助相似三角形原理，计算最终红色区域占比为：

以上部分就解释清楚了*shoulder effect*和*flank effect*.

(这些部分需要用相似三角形原理慢慢推导，不过思路基本类似，原理不难)

在整个算法中，基于以上推导进行了处理：





整个算法中对此部分进行了修正。