**Genetic Location Identification via Bionano**

**Version 1.6**

**User manual**

**Chunlong-Chen , Weitao Wang**

**2019-12-11**

**Contents**

1 Prerequisites

1. JDK version 1.8
2. R is required for peak calling
3. About 7GB of available RAM
4. Steps in this document assumes that your current working directory is ‘example’ and that the ‘replicon’ executable is in the parent directory.

2 Introduction

This is an extension software based on high throughput Bionano platform for the genomic position determination of any biological process correlated with fluorescently labeled DNA. (For example: DNA replication, DNA recombination, DNA repair) and provide the frequency occurred in population-based data for each found genomic position.

1. Clustering signal and get segmentation

The input data ask for Bionano [bnx](https://bionanogenomics.com/wp-content/uploads/2018/04/30038-BNX-File-Format-Specification-Sheet.pdf) and [xmap](https://bionanogenomics.com/wp-content/uploads/2017/03/30040-XMAP-File-Format-Specification-Sheet.pdf) dataformat files as most basic raw data. All analysis is based on these 2 files. The jar package should be run under linux terminal. The Patent\_GetSegment.jar can cluster neighbored signal based on the distance cutoff value set by user and merge the neighbored clustered primary segmentation. The default cluster cutoff value is 16384 and merged cutoff value is 32768. If you don’t want to merge the clustered primary segmentation to final segmentation, you can set the -MergeCutoff value as -1.

The must have parameter is -BNX, -XMP, -o, -FN and -Signal\_C

-BNX: The parent path to BNX file (The document where .bnx file exist)

-XMP: The parent path to XMP file (The document where .xmap file exist)

-o: The path of output file

-FN: The sample name list under the BNX and XMP path (Please pay attention the .bnx and .xmap must be with same file name beside suffix .bnx and .xmap, and must be end with such kind of suffix for dataformat recognition)

-Signal\_C: The bionano channel of signal in your experiment corresponding to mapping channel.

**The output is like below**

1. A .txt file record all detailed information about mapped fiber with red signal including red signal position, signal intensity and SNR.
2. A fiber coordinate bed file recorded all mapped fiber
3. A segmentation bed file recorded all final segmentation

**The other option**

-FL: You will get the mapped or unmapped DNA fiber length

-SignalPerSeg: pick up the segmentation which containing at least signal number you set here

-SignalPerFiber: pick up the fiber which containing at least signal number you set here

-FiberWithRed: filtering the fiber without red signal

-TXT: If you second try to modify any parameter, you can use the -TXT to set the .txt file generated by previous running.

**The example (Quick start)**

Java -jar Patent\_GetSegment.jar -BNX xxx/PathToBNX -XMP xxx/ PathToXMP -FN 1905async 1905\_FC1 1905\_FC2 -o xxx/test -Signal\_C 2 -FL -FiberWithRed -SignalPerSeg 3

Note: Under path to BNX must have 3 document named as 1905async.bnx , 1905\_FC1.bnx , 1905\_FC2.bnx. Same as path to XMP with 1905async.xmap , 1905\_FC1.xmap , 1905\_FC2.xmap

1. Count the signal number in each bin

The input data needs .txt file got by Patent\_GetSegment.jar. It used to generate bin files along the genome and count the signal number in each bin. The must have parameter is -WindowSize, -TXT, -Chr\_Length, -o, -Mode

-WindowSize: The bin width you set

-TXT: The path to file generated by Patent\_GetSegment.jar

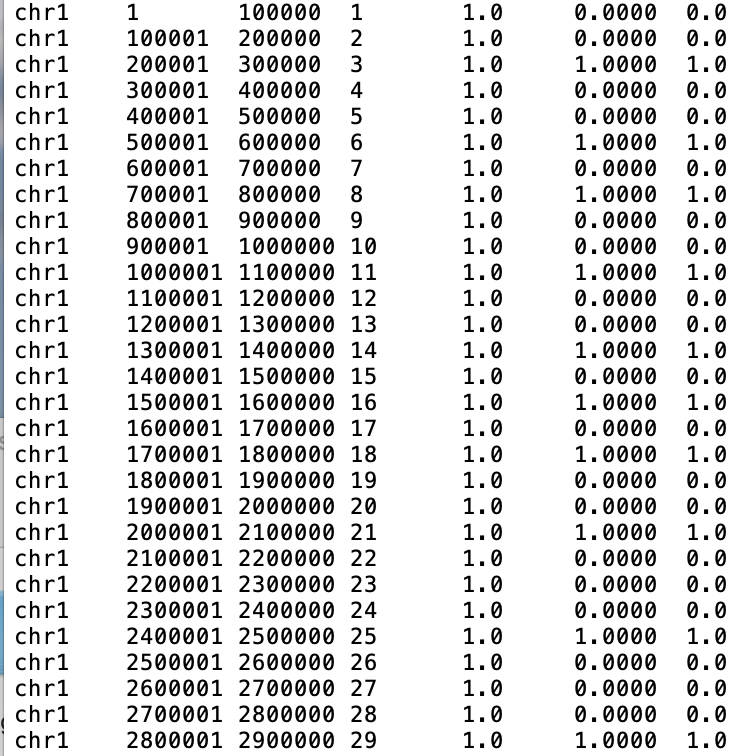
-Chr\_Length: The chromosome length (Refer to [UCSC](https://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/hg19.chrom.sizes))

-o: The path of output file

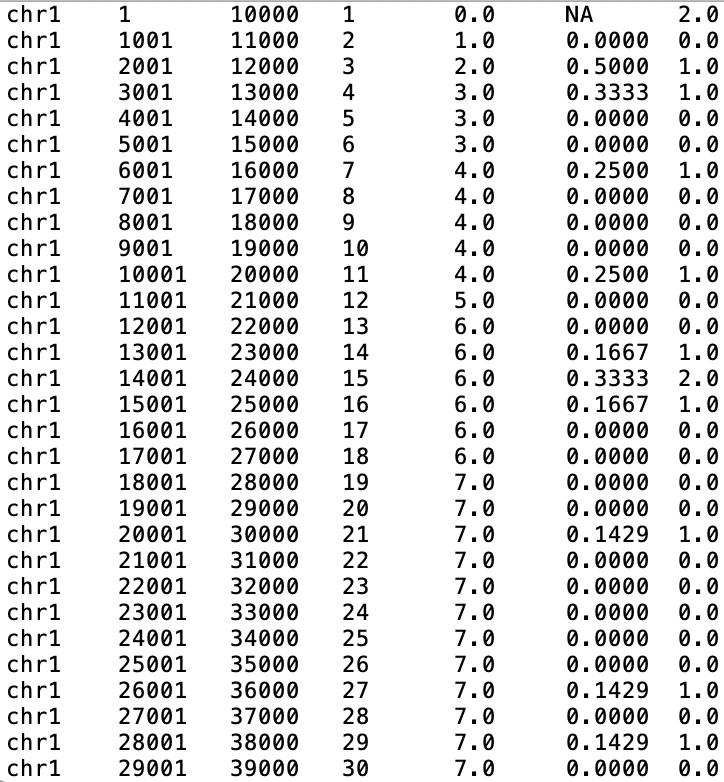
-Mode: (Adjacent/Sliding) the type of bin windows you want

**The output is like below**

Adjacent Bin (window\_size=100000, no overlapped adjacent window)



Sliding Bin (window\_size=10000, SlidingStep=1000)



**The other option**

-StepLength: When mode is Sliding you must set sliding step

-Normalize\_ByFiber: The path to xxx\_fiber.bed file generated by Patent\_GetSegment.jar used to normalize the signal by mapped fiber depth in local bins

**The example (Quick start)**

Java -jar Patent\_GetSegment.jar **-WindowSize 10000 -StepLength 1000**

**-TXT /Users/wwang/Desktop/test/1905async.txt**

**-o /Users/wwang/Desktop/test/**

**-Mode Sliding**

**-Chr\_Length /Users/wwang/Desktop/enrichment/ChrLength\_hg19.txt**

**-Normalize\_ByFiber /Users/wwang/Desktop/test/1905async\_fiber.bed**

1. Generate GTF

The input data needs .txt file got by AllRawDataRefining\_1R2G.jar or AllRawDataRefining\_1G2R.jar. It used to generate gtf like files to record genomic positions of all red signals and ends position of fiber. The must have parameter is -I, -O,-SampleName

-I: The input txt file’s parent path

-O: The output gtf file’s parent path

-SampleName: the name of all txt file, multi files will be separated by “,” and the name of txt file can’t contain character “\_” since it has been used as delimiter in script.

-Contain: The option could only be T when it show up, the default value is false when it is missing. It used to filter out the fiber without signal.

**The example (Quick start)**

java -jar GenerateGTF\_ByAllDataRefining\_Reformat.jar

-I /Volumes/WWT/Reformat/TXT/1802/

-O /Users/wwang/Desktop/ORM/

-SampleName 1802.0a.txt,1802.0b.txt

-Contain T

Notice: The red option could be removed based on clients’ requirement.

1. GetFiberCoordinate\_ByTXT

The input data needs .txt file got by AllRawDataRefining\_1R2G.jar or AllRawDataRefining\_1G2R.jar. It used to generate fiber’s bed files. The must have parameter is -I, -O,-SampleName

-I: The input txt file’s parent path

-O: The output file’s parent path

-SampleName: the name of all txt file, multi files will be separated by “,” and the name of txt file can’t contain character “\_” since it has been used as delimiter in script.

**The example (Quick start)**

java -jar GetFiberCoordinate\_ByTXT.jar

-I /Volumes/WWT/Final-Version/1807\_1802\_1905\_1708\_Merge/TXT/AlldataRefining/0min/

-O /Users/wwang/Desktop/ORM/GetFiberCoordinate\_ByTXT/Test

-SampleName 1905.FC0.txt,1905.FC1.txt

1. GetRedflagNumberInSlidingWindow\_ByAllDataRefining

The input data needs .txt file got by AllRawDataRefining\_1R2G.jar or AllRawDataRefining\_1G2R.jar. It used to generate fiber’s bed files. The must have parameter is -I, -O, -SampleName

-I: The input txt file’s parent path

-O: The output file’s parent path

-SampleName: the name of all txt file, multi files will be separated by “,” and the name of txt file can’t contain character “\_” since it has been used as delimiter in script.

-REF: The input REF file Unit kb, the example like below

chr1 249250621

chr2 243199373

chr3 198022430

…

Please pay attention, here the chromosome name must be same as the chromosome names show up in .txt files even including uppercase and lowercase letters Chr and chr is different, sometrimes people use chr23 and chr24 to represent chrX and chrY in .txt file. So corresponding chromosome name in this REF file should be chr23 and chr 24 instead of chrX and chrY

-WS: window size, Unit kb

-SS: SlidingStep, Unit kb

**The example (Quick start)**

java -jar GetRedflagNumberInSlidingWindow\_ByAllDataRefining.jar

-REF /Users/wwang/Desktop/enrichment/ChrLength\_hg19.txt

-WS 10

-SS 1

-I /Volumes/WWT/Final-Version/1905/TXT/AllDataRefining/

-SampleName 1905.FC1.txt,1905.FC2.txt

-O /Users/wwang/Desktop/ORM/GetRedflagNumberInSlidingWindow\_ByAllDataRefining/Test

**Output format**

Chr Bin\_start Bin\_end Bin\_ID RedsignalNumber FiberNumber FireEfficiency FiberwithoutSignal  
  
FireEfficiency = RedsignalNumber / FiberNumber , it is a value after fiber depth normalization to estimate signal showing up probability

1. GetRedflagNumberInAdjacentWindow\_ByAllDataRefining

The input data needs .txt file got by AllRawDataRefining\_1R2G.jar or AllRawDataRefining\_1G2R.jar. It used to generate fiber’s bed files. The must have parameter is -I, -O,-SampleName

-I: The input txt file’s parent path

-O: The output file’s parent path

-SampleName: the name of all txt file, multi files will be separated by “,” and the name of txt file can’t contain character “\_” since it has been used as delimiter in script.

-REF: The input REF file Unit kb, the example like below

chr1 249250621

chr2 243199373

chr3 198022430

…

Please pay attention, here the chromosome name must be same as the chromosome names show up in .txt files even including uppercase and lowercase letters Chr and chr is different, sometrimes people use chr23 and chr24 to represent chrX and chrY in .txt file. So corresponding chromosome name in this REF file should be chr23 and chr 24 instead of chrX and chrY

-WS: window size, Unit kb

**The example (Quick start)**

java -jar GetRedflagNumberInSlidingWindow\_ByAllDataRefining.jar

-REF /Users/wwang/Desktop/enrichment/ChrLength\_hg19.txt

-WS 10

-I /Volumes/WWT/Final-Version/1905/TXT/AllDataRefining/

-SampleName 1905.FC1.txt,1905.FC2.txt

-O /Users/wwang/Desktop/ORM/GetRedflagNumberInSlidingWindow\_ByAllDataRefining/Test

**Output format**

Chr Bin\_start Bin\_end Bin\_ID RedsignalNumber FiberNumber FireEfficiency FiberwithoutSignal  
  
FireEfficiency = RedsignalNumber / FiberNumber , it is a value after fiber depth normalization to estimate signal showing up probability

1. GetNewSegmentation\_AddSoloSignal

-I: The input txt file’s parent path

-O: The output file’s parent path

-SampleName: the name of all txt file, multi files will be separated by “,” and the name of txt file can’t contain character “\_” since it has been used as delimiter in script.

-GetSignalDistance: if it is “T” the script will generate a document record all the signal internal distance, the default value is false.

-FC: The first cutoff values to cluster the closet signal. Multi cutoff values for different files could be separate by “,” User point a cutoff value for internal signal distance and all signals within this value will cluster into segemtation

-GetSegmentDistance: if it is “T” the script will generate a document record all the signal internal distance, the default value is false. And you should put the -FC options in front of -GetSegmentDistance. Because -FC is a must have options for GetSegmentDistance.

-SC: The second cutoff values to cluster the closet segmentation. Multi cutoff values for different files could be separate by “,” User point a cutoff value for internal segmentation distance and all segmentations within this value will cluster into one segmentation and the script will generate the final bed files

"-AddSolo" : If it is “T”, the final result will contain all single signal coordinate without clustering. The default is false;

**The example (Quick start)**

java -jar GetNewSegmentation\_AddSoloSignal.jar

-GetSignalDistance

-I /Volumes/WWT/Final-Version/1905/TXT/AllDataRefining/

-SampleName 1905.FC1.txt,1905.FC2.txt

-O /Users/wwang/Desktop/ORM/GetNewSegmentation\_AddSoloSignal/Test

java -jar GetNewSegmentation\_AddSoloSignal.jar

-FC 16384,16384

-GetSegmentDistance

-I /Volumes/WWT/Final-Version/1905/TXT/AllDataRefining/

-SampleName 1905.FC1.txt,1905.FC2.txt

-O /Users/wwang/Desktop/ORM/GetNewSegmentation\_AddSoloSignal/Test

java -jar GetNewSegmentation\_AddSoloSignal.jar

-FC 16384,16384

-SC 32768,32768

-I /Volumes/WWT/Final-Version/1905/TXT/AllDataRefining/

-SampleName 1905.FC1.txt,1905.FC2.txt

-O /Users/wwang/Desktop/ORM/GetNewSegmentation\_AddSoloSignal/Test

(-AddSolo T)