Supplemental Information 2

Readme.txt

- RefAssemblyChIP.sh: Using this script, trimmed ChIP paired-end reads in FASTQ format are aligned to the reference *P. berghei* ANKA genome. Commands that '.bam' files are produced and converted to '.bed' files using the SAMtools 'bamtobed' command.
- runbamCompare.sh: The bamCompare command aligns trimmed and sorted reads to a specified control/input and outputs a '.bed' file.
- runbamCompare2_bigwig.sh: Converts the '.bed' file from 'bamCompare' to a '.bigwig' file that can be viewed in a genome browser.
- FindPeaksEpic.sh: Commands that the Epic program be used to find ChIP peaks. The reference genome (in FASTA format) and, control/input BED file, and an aligned sample/control BED file are used as input here. The output is a 'modified.bed' file.
- runEpic.sh: Script in which peaks called using Epic ('modified.bed') are compared to an input peak file for comparative analysis. Output is an 'epicout.bed' file.
- format_epic_for_annotation.sh: Prepares Epic output files for annotation using the AnnotatePeaks.pl script (modified HOMER script in Perl programming language) by formatting the 'epicout.bed' file correctly. Output is a 'formatted.bed' file.
- runAnnotatePeaks.sh: Peaks that were called using Epic are annotated using data from the '.gff' file of the *P. berghei* ANKA reference genome (in FASTA format) from *Plasmo*DB version 34 with the AnnotatePeaks.pl script (modified HOMER script in Perl programming language). The input is a 'formatted.bed' file and the output is an 'annotated.bed' file.

RefAssemblyChIP.sh

```
#!/bin/bash
```

```
# @author Sejal Modha
# Run the script using 4 params
# $1=read_f1.fq $2=read_f2.fq $3=reference.fa $4=OutPrefix

usage=`echo -e "\n Usage: RefAssemblyChIP.sh file_R1.fastq
file_R2.fastq bowtie2Indexes OutputPrefix \n"`;

if [[ ! $1 ]]
then
        printf "${usage}\n\n";
exit;
fi

echo "Processing Input files" $1 $2

file1=$1
file2=$2
file3=$3
#echo $file1;
#echo $file2;

if [[ $file1 == *.fastq || $file1 == *.fq ]]
        then
        f1="${file1%.*}";
fi
echo $f1;

if [[ $file2 == *.fastq || $file2 == *.fq ]]
        then
        f2="${file2%.*}";
fi
echo $f2;
```

```
if [[ $file3 == *.fa || $file2 == *.fasta ]]
        then
        f3="${file3%.*}";
fi

echo "Reference genome set to" $f3;

trim_galore --paired $1 $2

mv ${f1}_val_1.fq ${4}_val_1.fq
mv ${f2}_val_2.fq ${4}_val_2.fq

echo "Running Bowtie2";

#bowtie2-build $3 $f3;

bowtie2 -x $3 -1 ${4}_val_1.fq -2 ${4}_val_2.fq -S ${4}_bt2.sam
-p 8

samtools view -bh -@ 8 -S ${4}_bt2.sam -o ${4}_bt2.bam

samtools sort -@ 8 -n -o ${4}_bt2_sorted_name.bam ${4}_bt2.bam

bamToBed -bedpe -i ${4}_bt2_sorted_name.bam >
${4}_bt2_sorted_name.bed

rm $4*.sam ${4}_bt2.bam
```

runbamCompare.sh

```
for i in `ls *.bam`
do
    echo $i
    samtools sort -@ 10 -o ${i}_sorted.bam $i
    samtools index ${i}_sorted.bam
    bamCompare -b1 ${i}_sorted.bam -b2
S3_INPUT_250k_PbANKA_bt2_sorted.bam  -o
${i}_INPUT_bamcompared.bed -of bedgraph -bs 2000
done
```

runbamCompare2_bigwig.sh

```
for i in `ls *_sorted.bam`
do
    echo $i
    #samtools sort -@ 10 -o ${i}_sorted.bam $i
    #samtools index ${i}_sorted.bam
    bamCompare -b1 ${i} -b2 S3_INPUT_250k_PbANKA_bt2_sorted.bam
-o ${i}_INPUT_bamcompared.bigwig
done
```

FindPeaksEpic.sh

```
#!/bin/bash
#@author Sejal Modha
# Run the script using 4 params
# $1=reference_genome.fa $2=ReadLength $3control_sample.bed
$4=treated_sample.bed $5=OutPrefix
```

```
usage=`echo -e "\n Usage: FindPeaksEpic.sh ReferenceGenome.fasta
ReadLength ControlSample.bed TreatedSample.bed OutputPrefix"`;

if [[ ! $1 ]]
then
        printf "${usage}\n\n";
exit;
fi

echo ""
echo -e "Processing Input files" $1 $3 $4

file1=$1
file3=$3
file4=$4
#echo $file1;
#echo $file2;

if [[ $file1 == *.fasta || $file1 == *.fa ]]
        then
        f1="${file1%.*}";
else
    echo ""
    echo "Please input reference genome in the fasta format"
    echo "${usage}";
    echo ""
    exit;
fi
#echo -e $f1;

if [[ $file3 == *.bed || $file3 == *.bedpe ]]
```

```
        then
        f3="${file3%.*}";
fi
#echo -e $f3;

if [[ $file4 == *.bed || $file4 == *.bedpe ]]
        then
        f4="${file4%.*}";
fi

echo -e "Reference genome set to" $f1;
echo -e "Control file is" $file3
echo -e "Treated file is" $file4

echo
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"
echo -e "Calculating Effective Genome Size for the reference"
echo
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"

epic-effective -r $2 $1 > ${5}_epic_effective_gs
eff_gs=`grep "Effective genome size" ${5}_epic_effective_gs|cut
-f2 -d":"|sed 's/ //g'`
echo $eff_gs

echo
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"
echo -e "Generating chrsizes file"
```

```
echo
"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"
samtools faidx $1
cut -f1-2 ${1}.fai > ${5}_chrsizes
echo -e "Done"
echo
"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"
echo -e "Calling peaks using epic - this might take a while"
echo
"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++"
epic -t $4 -c $3  -cs ${5}_chrsizes -egf $eff_gs -o ${5}_epicout
-cpu 2 -l ${5}_epic.log -b ${5}_epicout.bed

awk -F"\t" '{print $1"\t"$2"\t"$3"\t""Peak_"NR"\t"$4"\t"$5}'
${5}_epicout.bed > ${5}_epicout_modified.bed
echo -e "epic output file: "${5}_epicout
echo -e "epic output in .bed format file: "${5}_epicout.bed
echo -e "epic run log file: "${5}_epic.log
echo -e "Edited .bed file: "${5}_epicout_modified.bed
```

runEpic.sh

```
for i in `ls *.bed`
do
    echo $i
    epic -t $i -c S3_INPUT_250k_PbANKA_bt2_sorted_name.bed  -cs
../Ref/PbergheiANKA.chrsizes -egf 0.963252942956 -o
${i}_INPUT_epicout -cpu 8 -l ${i}_epic.log -b
${i}_INPUT_epicout.bed
```

```
done
```

format_epic_for_annotation.sh

```
for i in `ls *_epic`
do
    echo $i
    awk -F"\t" '{print $1"\t"$2"\t"$3"\t"$4"\t"$5"\t""+""\t"$6}'
$i > ${i}_formatted.bed
done
```

runAnnotatePeaks.sh

```
for i in `ls *.bed`
do
    echo $i
    filename="${i%.*}"
    annotatePeaks.pl $i ../Ref/PlasmoDB-
34_PbergheiANKA_Genome.fasta -gff3 ../Ref/PlasmoDB-
34_PbergheiANKA.gff > ${filename}_annotated.bed
done
```