

# A User's Guide for PSCN

Hao Chen

April 13, 2011

PSCN is an R package which gives an estimation of parent-specific DNA copy number of human genomes based on high-density SNP arrays data, that is, the data has two component, logR and B-allele frequency. This package can also be applied on dataset having both SNP probes and copy number probes.

## 1 Input data format

PSCN is applied to text file with the following format:

Position	Chr	logR	bfreq
38411	3	0.07381305	0.47021
41894	3	-0.1297997	0.9724417
57010	3	-0.1671092	0.004669038
70973	3	0.2277164	0.4676426
79972	3	-0.05290581	0.4790407
82626	3	0.03966476	0.002807678
...			

Or one more column containing information of whether a probe is a SNP probe or a copy number probe.

Position	Chr	logR	bfreq	probe
527	17	0.13937900	0.0000000000	CN
3549	17	-0.20710826	0.0000000000	CN
3887	17	0.02859726	0.0000000000	CN
6689	17	0.07442529	0.9370690526	SNP
6888	17	0.35640816	0.0007204988	SNP
10268	17	-0.10509859	0.0000000000	CN

## 2 Inference

In this package, three functions are called to do the inference.

“smoothing”: estimate parent-specific DNA copy number for each position by maximizing

likelihood.

“segmentation”: determine change points.

“pscnlist”: generate lists showing major and minor copy numbers.

In the following, the chromosome 3 from sample TCGA.02.0332 is used as an example.

```
## install package and load data
install.packages("PSCN")
library(PSCN)
data(Illu0332chr3)

## TCGA.02.0332 Chromosome 3 from the Illumina platform
smoothing(samplename="TCGA.02.0332", inputdata=Illu0332chr3, platform="Illumina")
segmentation(samplename="TCGA.02.0332", chrs=3, MIN.SNPS=50)
pscnlist(samplename="TCGA.02.0332", chrs=3, MIN.SNPS=50)
```

The output files of the function “pscnlist” are of names: samplename.longlist.txt, samplename.shortlist.txt, samplename.shortlist2.txt. The samplename.shortlist2.txt is output only when the argument GLBalance is TRUE. The difference between the samplename.longlist.txt and samplename.shortlist.txt is that in the shortlist, the consecutive segments with the same type of change status, such as both are Gain/Loss, are combined.

For dataset having both SNP and copy number probes, we pre-specify the prior distribution of genotype configuration of copy number probes since the DNA sequence is the same on both chromosomes.

```
data(AffySW1417chr17)

# prepare genotype frequency for copy number probes
n = dim(AffySW1417chr17)[1]
genotype.freq = matrix(NA, n, 4)
for (i in 1:n){
  if (AffySW1417chr17$probe[i] == "CN"){
    genotype.freq[i,] = c(1,0,0,0)
  }
}

smoothing(samplename="SW1417", inputdata=AffySW1417chr17,
  genotype.freq=genotype.freq, platform="Affymetrix")
segmentation(samplename="SW1417", chrs=17, regroup.percent=0.15)
pscnlist(samplename="SW1417", chrs=17)
```

### 3 Visualizing Results

“pscn.plot” is the plot function to visualize the results.

We can use this function to view major and minor copy number.

```
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="copy number")
# if we want x axis to be position rather than SNP index and show title
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="copy number",
use.pos=TRUE, use.main=TRUE)
# change gain, loss, normal colors
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="copy number",
use.pos=TRUE, use.main=TRUE, col.gain="purple", col.loss="yellow",
col.normal="lightblue")
```

we can use this function to view logR, Bfrequency, allele A raw copy number, allele B raw copy number, or total raw copy number (R).

```
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq")
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="logR")
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="A")
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="B")
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="R")

# if do not want to see the black vertical lines showing changepoint
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
changepoint=FALSE)
# to color different segment differently
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq", region=TRUE)
# only view SNPs from 20000 to 30000
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
loc=20000:30000)
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
changepoint=FALSE, loc=20000:30000)
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
region=TRUE, loc=20000:30000)
# only view segments 4-8
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
region=TRUE, regionid=4:8)
pscn.plot(samplename="TCGA.02.0332", chrid=3, which.plot="bfreq",
region=TRUE, regionid=4:8, region.col=c("blue","green","yellow","red","black"))
#specifies the colors for segments by oneself
```

We can use this function to view scatter plots and contour plots, which are plotting B allele intensity vs. A allele intensity.

```
# Scatter plot
pscn.plot(samplename="TCGA.02.0332", chrid=3, scatter=TRUE)
pscn.plot(samplename="TCGA.02.0332", chrid=3, scatter=TRUE, loc=20000:30000)
pscn.plot(samplename="TCGA.02.0332", chrid=3, scatter=TRUE, regionid=3)

# contour plot
pscn.plot(samplename="TCGA.02.0332", chrid=3, Contour=TRUE)
pscn.plot(samplename="TCGA.02.0332", chrid=3, Contour=TRUE, loc=20000:30000)
pscn.plot(samplename="TCGA.02.0332", chrid=3, Contour=TRUE, regionid=2)
pscn.plot(samplename="TCGA.02.0332", chrid=3, Contour=TRUE, regionid=2:3)
```