DK: A Package for detecting DNA copy amplifications by Digital Karyotyping

Chao-Ling Chang, Leslie Cope, Giovanni Parmigiani

April 25, 2006

Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Contents

1 Introduction

Digital Karyotyping (`DK`) is a method that provides quantitative analysis of DNA copy number at high resolution. DK uses cleaving enzymes to obtain unique tags (short but representative DNA sequences) at approximately 4 kb intervals along the entire genome. The DNA tags are amplified, and then identified and counted by sequencing and tallied. Many copies indicate amplification, while few copies suggest a deletion. A full description of the DK approach, detailed protocol for DK and sample data are available at the DK website : `http://www.digitalkaryotyping.org/`.

The goal of DK data analysis is to identify genomic regions in which tag counts are consistently higher or lower than expected by chance alone. In this project we concentrate on amplifications, but the basic ideas extend to deletions as well. The current approach to identifying the amplifications is to average the gene specific DNA copy numbers in a sliding window, comparing results to a null distribution obtained by permutation. Here, we save the computational expense of the simulations, and improve analysis by introducing a formal framework for statistical inference from DK data.

Because our data consists of tag counts, it would be natural to consider the Poisson distribution for our model and in fact, the permutation null distribution can be completely described in terms of Poisson processes. Here we develop an **R** package for the analysis of DK data based on Poisson distribution assumption. In this package, we also have to consider the data structure for storage of DK data. Since there will be different chromosomes, different libraries in DK data files, how to use minimal space to contain all the information, and to classify different types of information will be our main concern. The existing **eSet**, which is a general container for high-throughput assays and experimental metadata in `Bioconductor`, is fit for the need of our data and used as data structure template. In addition, in our R package, formulas for the calculation of power and sample size are also provided. Finally there are four implementing functions, `readDK`, `processDK`, `plotDK` and `powerDK` in this R package. The descriptions of the four functions are on the chapter 3.

2 Data

Data used in this package was generated in the lab of Dr. Tian-Li Wang at JHMI Sidney Kimmel Comprehensive Cancer Center. In DK data files, there are six columns, they are: (1) GTAG: Gene Tags (2) Chr: Chromosome number (3) Pos: Position in chromosome (4) Orient: orientation in chromosome (5) Dist: Distance (6) Count: Number of tag counts

The data format can be found at
`http://www.biostat.jhsph.edu/~clchang/project/content.html#data`

3 Examples

3.1 Create object of class "DK" from digital karyotyping files

Here we demonstrate reading in digital karyotyping files and combining them into one DK object, which has the same slots as class **eSet**. We use the slot, asssayData, to subtract the tag counts information.

```
> library(DK)
> datadir <- system.file("data", package = "DK")
> dk.obj <- readDK(filenames = listDKfiles(datadir), sep = "\t",
+     phenoData = NULL, description = NULL, notes = "", sampleNames = NULL)

Trying to read /home/bst/student/clchang/Rlibs/DK/data/LIB1.TXT
Trying to read /home/bst/student/clchang/Rlibs/DK/data/LIB2.TXT

> dk.obj

instance of eSet
assayData component is of class list
dimensions of the assayData components:
     tagCounts
nrow      6000
ncol         2
         phenoData object with 1 variables and 2 cases
         varLabels
                : sample
first reporterNames:
[1] "cgctgctccaccttcgg" "cctctctgggcctgcgc" "atgggaccccgcgcagg"
[4] "ccgccactatactgtgt" "tagaaagggaagacata"
first sampleNames:
[1] "/home/bst/student/clchang/Rlibs/DK/data/LIB1.TXT"
[2] "/home/bst/student/clchang/Rlibs/DK/data/LIB2.TXT"
[3] NA
[4] NA
[5] NA
```

3.2 Process Poisson Tests

Here we process Poisson test to each sliding windows, which contains specific length of tag counts. As for how to choose the appropriate window size,

please see section 4.4. Then, the function combines the results and additional window sizes information, which is stored in the "history" slot, to original DK object and return the new processed DK object.

```
> data(dk.example)
> dk.proc <- processDK(dk.example, win.size = 50)
> stats <- assayData(dk.proc)[[2]]
> p.vals <- assayData(dk.proc)[[3]]
> dk.proc

instance of eSet
assayData component is of class list
dimensions of the assayData components:
     tagCounts stats p.values
nrow      6000  6000     6000
ncol         2     2        2
         phenoData object with 1 variables and 2 cases
         varLabels
                : sample
first reporterNames:
[1] "cgctgctccaccttcgg" "cctctctgggcctgcgc" "atgggaccccgcgcagg"
[4] "ccgccactatactgtgt" "tagaaagggaagacata"
first sampleNames:
[1] "./lib1.txt" "./lib2.txt" NA           NA           NA
```

3.3 Plot for the tag counts, fold changes and the amplification detections

When processing the plot, you have to specify which sample and which chromosome is your target. If the input DK object contains only raw data, then it will only plot the tag counts. If the input DK object has been processed, then it will plot tag counts, fold changes and amplification detections.

```
> plotDK(dk.proc, sample.ind = 2, chr = 4)
```

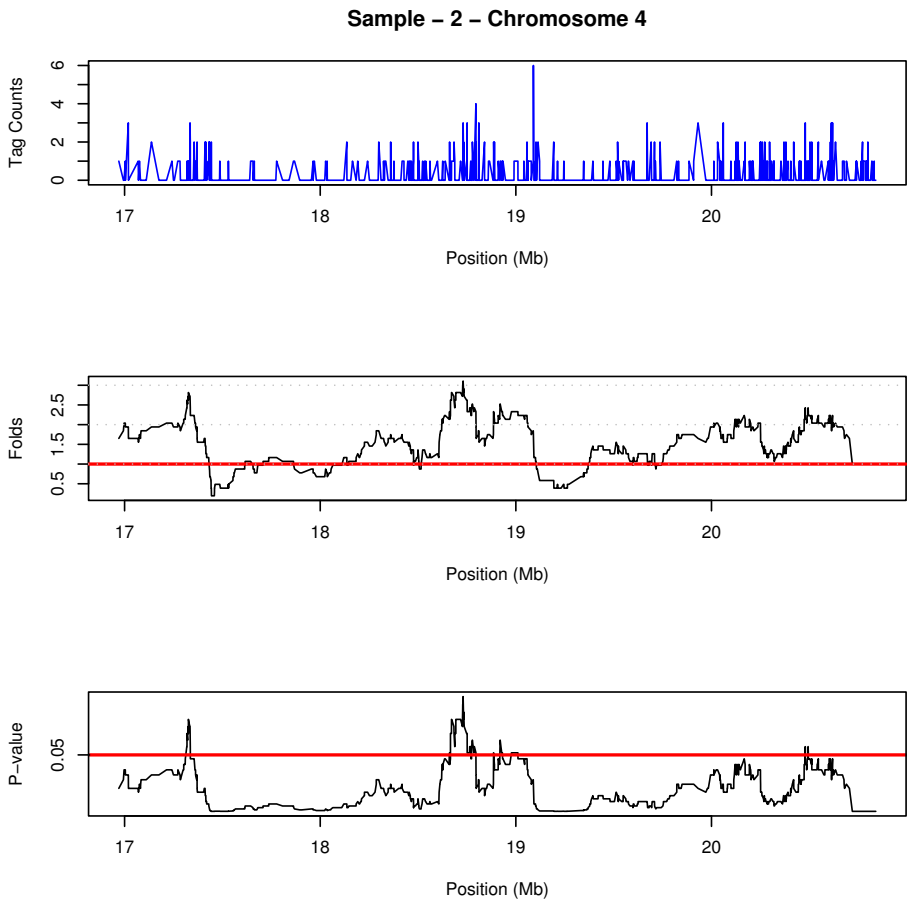

Figure 1: Show the tag counts, fold changes, and the amplification detection

3.4 Power calculations for one and multiple libraries/samples

We have developed two functions here to do the power calculations. The first function is **onelib.power.test**. This function permits various power and sample size calculations within one library. You can leave one parameter as `NULL`, but only one can be `NULL`, then the **onelib.power.test** can answer it for you. Since the adjusted p-value depends on the number of virtual tags

in one chromosome, so it is included as an argument, which is presented as "length". Foe example, we want to know the window size used in scan test, and then we can use the function as:

```
> onelib.power.test(win.size = NULL, fold.crit = 2, rate = 0.15,
+     power = 0.8, length = 40000)

     Power Calculation for one library

       win.size = 220
      fold.crit = 2
           rate = 0.15
          alpha = 0.05
          power = 0.8
```

The function **multilib.power.test** permits power and sample size calculations concerning the number of libraries. Similarly, we can leave the interested parameter as NULL, but only one parameter can be NULL, and the multilib.power.test will answer it for you. For example, we want to calculate the number of libraries, we use the function as:

```
> multilib.power.test(lib.num = NULL, onelib.power = 0.6, detect.rate = 0.85,
+     mut.rate = 0.2)

     Power Calculation for multiple libraries

        lib.num = 15
   onelib.power = 0.6
    detect.rate = 0.85
```

4 Acknowledgements

The authors would like to express their gratitude to Dr. Tian-Li Wang for sharing the data. Many thanks to Rob Scharpf and Benilton Carvalho for assistance in package-build debugging.

References

1. Tian-Li Wang, Christine Maierhofer, Michael R. Speicher, Christoph Lengauer , Bert Vogelstein , Kenneth W. Kinzler , and Victor E. Velculescu. Digital Karyotyping. Proc Natl Acad Sci USA, Vol. 99, no. 25, 16156-16161, 2002.

2. Tian-Li Wang, Diaz LA Jr, Romans K, Bardelli A, Saha S, Galizia G, Choti M, Donehower R, Parmigiani G, Shih IeM, Iacobuzio-Donahue C, Kinzler KW, Vogelstein B, Lengauer C, Velculescu VE. Digital karyotyping identifies thymidylate synthase amplification as a mechanism of resistance to 5-fluorouracil in metastatic colorectal cancer patients. Proc Natl Acad Sci US A. 101(9):3089-94, 2004.

3. http://www.biostat.jhsph.edu/ clchang/project/index.html