CIMPL Vignette

Jelle ten Hoeve and Jeroen de Ridder

November 23, 2009

1 Introduction

In this Vignette we demonstrate the usage of the `cimpl` R package. CIMPL is the abbreviation for Common Insertion site Mapping PLatfrom. This package identifies common insertion sites (CISs) given insertional mutagenesis data collected across a cohort of tumors. The implementation is based on the Gaussian Kernel Convolution (GKC) framework developed by de Ridder *et al.* Originally, this method was developed for analysis of retroviral insertional mutagenesis data. However, we have extended the method to enable the analysis of transposon insertional mutagenesis as well. In cancer research, well known used retro-viruses are Mouse Mammary Tumor Virus (MMTV) and Murine Leukemia Virus (MuLV). Sleeping Beauty (SB) and piggyBac (PB) are examples of transposon-based systems. In this Vignette we walk through the steps for calculation and visualization of the results.

2 Data

First, we load an insertional mutagenesis dataset. The dataset has to be a `data.frame` containing mapped insertion locations. Columns named 'chr' (chromosome) and 'location' are mandatory. Not required (but useful) columns are 'sampleID' if you want to plot tumor densities (Section 3.2) and 'contig_depth' if local hopping correction is needed. All other columns are not used. However, they will be reported in the results (Section 4) and thus, it might be useful to include other columns for one's own further analysis.

In this package, we have included data of a transposon based screen in mice developing colorectal cancer [2]. This data was harvested from 135 tumors containing 16690 non-redundant mapped insertions. As can be seen, some additional columns are included.

```
> library(cimpl)
> data(colorectal)
> str(colorectal)

'data.frame':        16690 obs. of  8 variables:
 $ sampleID      : chr  "S357" "S487" "S357" "S357" ...
 $ tissueType    : chr  "Int" "Int" "Int" "Int" ...
 $ tissueID      : chr  "A" "C" "D" "A" ...
 $ chr           : chr  "chr1" "chr1" "chr1" "chr1" ...
 $ location      : int  6053912 8355383 16607442 17511946 22556961 22914215 23907046 28285
 $ associatedGene: chr  "" "" "" "" ...
 $ cohort        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ contig_depth  : num  1 1 1 1 1 1 1 1 1 1 ...
```

Next, we load the BSgenome on which the insertions are mapped. Make sure you use exactly the same genome version as used for the mapping! Package BSgenome.Mmusculus.UCSC.mm9 provides a Mmusculus object (which we use later) and is based on the NCBI m37 mouse assembly, which is compatible with the colorectal data.

```
> library(BSgenome.Mmusculus.UCSC.mm9)
```

We are now ready to apply the GKC framework.

3 Analysis

In this example we ran only 100 iterations, selected only Chromosome 18 and varied the scale for only three values. This to minimize computation time. Normally, we recommend 10000 iterations to obtain appropriate p-values. Select all chromosomes, unless, for example, the experimental design forces you to leave out chromosomes that contain the donor site. Regarding scales, [1k, 2.5k, 5k, 10k, 30k, 50k, 100k, 150k] is recommended for retroviral data and [10k, 20k, 30k, 40k, 50k, 60k, 70k, 80k, 90k, 100k, 110k, 120k, 130k, 140k, 150k] for transposon data. If you are not interested in comparing results from different scales, a single scale of 30k for retroviral data is reasonable [1]. For transposon data, no clear guideline exists yet. The following command executes the CIMPL analysis.

```
> ca <- doCimplAnalysis(colorectal, scales = c(30000, 50000, 70000),
+     chromosomes = c("chr18"), n_iterations = 100, BSgenome = Mmusculus,
+     system = "SB", lhc.method = "none")

doCimplAnalysis(data = colorectal, scales = c(30000, 50000, 70000),
    n_iterations = 100, chromosomes = c("chr18"), BSgenome = Mmusculus,
    system = "SB", lhc.method = "none")
Starting CIMPL analysis...
>>>> chromosome chr18 <<<<
h = 30000, null-peaks..................................................................
h = 50000, null-peaks..................................................................
h = 70000, null-peaks..................................................................
```

The steps performed in doCimplAnalysis are explained below.

3.1 Null distribution

First, random data are generated by permuting the positions of the insertions in the dataset. For each chromosome and each scale, the insertions are randomly distributed across the genome. Although retro-viruses can integrate anywhere on the genome (not taking insertion biases into account), transposons are usually restricted to certain genomic sequences, called specificity patterns. For example, the SB transposon can only insert at TA sites. Therefore, such a restriction can

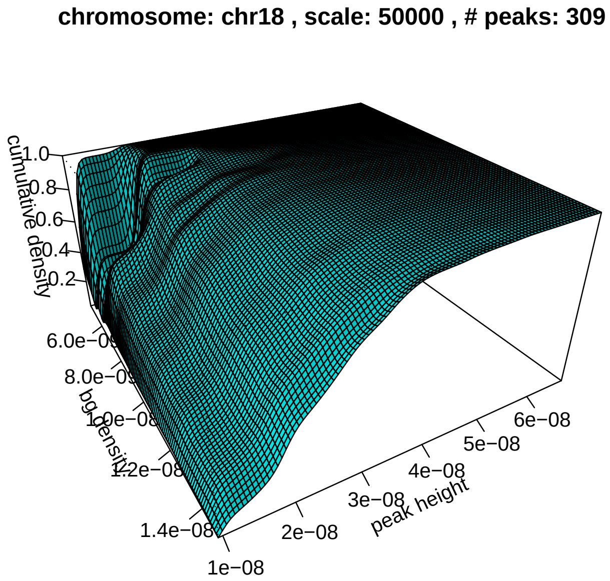

Figure 1: The null-distribution of peaks in terms of peak height and specificity pattern density.

be set using the `specificity.pattern` argument. When a restriction pattern is supplied, random insertion perturbations are only allowed on the site define by the specificity pattern. For convenience, you can set the `system` argument to MMTV, MuLV, SB or PB. The specificity pattern is then automatically set.

Secondly, on these random data GKC is performed and the peaks are detected. For each peak we save the peak height and the specificity pattern (or background) density. If no specificity pattern is used, the background distribution is uniform (`1/chr_length`), else, the background distribution is computed using GKC on the locations of the specificity pattern. This process of randomly permuting the data, performing GKC, and calculating the peak heights and local specificity pattern density (if applicable) is repeated many times. When a specificity pattern is provided, an empirical distribution of peak height conditioned on specificity pattern density can be constructed (Figure 1). When no specificity pattern is provided, a single empirical distribution function of peak height is constructed. The conditional distribution function is used for the significance calculation of peaks found in the observed data. More specifically, it allows the computation of the probability that a peak observed in the real data exceeds a peak height observed in the $\alpha\%$ of the random cases.

```
> plot(ca, type = "null.cdf", chr = "chr18", scale = 50000)
```

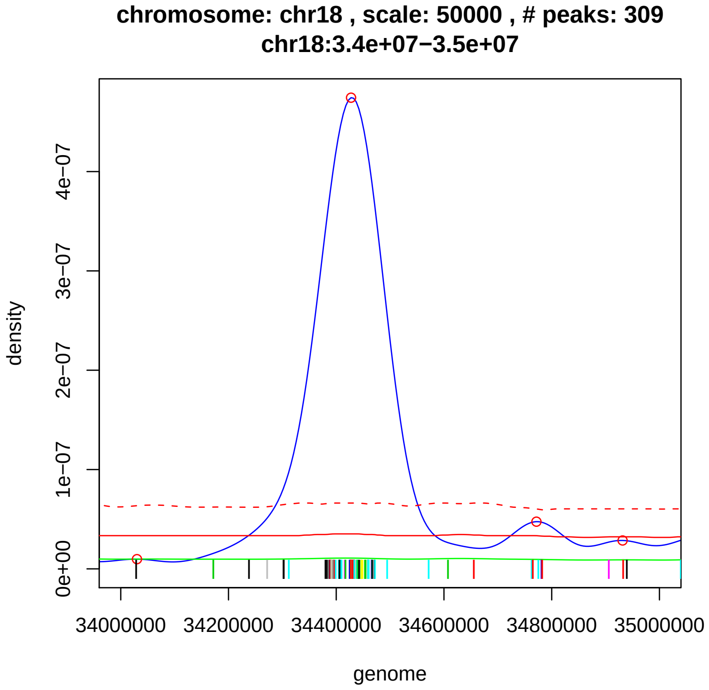

Figure 2: The kse plot.

3.2 Kernel smoothed estimate

Finally, GKC is performed on the observed data, resulting in a kernel smoothed estimate (kse) of the insertion density. The result can be visualized with the plot command:

```
> plot(ca, type = "kse", chr = "chr18", scale = 50000, bpLim = c(3.4e+07,
+     3.5e+07), interactive = FALSE)
```

Figure 2 gives details about Chromosome 18: 34Mb-35Mb for scale 50k, which contains a CIS. The CIS is the region where the kse (blue line) is above the significance threshold (red solid line) or above the significance threshold corrected for multiple testing (dotted red line). Peaks are indicated with red circles. The vertical lines at the bottom indicate the insertions color-coded by the sample identifiers (‘sampleID’ column in the data).

By default, the significance threshold is set to 5% and Bonferroni multiple testing correction is applied. This can be modified using the `alpha` and `mul.test` arguments. If you set `interactive=TRUE`, one can interactively browse over the genome using the mouse.

Additionally, a ‘lava’ plot can be constructed (Figure 3), where the colored bands (layers of lava) represent the contribution of each sample to the kse.

```
> plot(ca, type = "kse", chr = "chr18", scale = 70000, bpLim = c(3.4e+07,
+     3.5e+07), plot.tumor.densities = TRUE, interactive = FALSE)
```

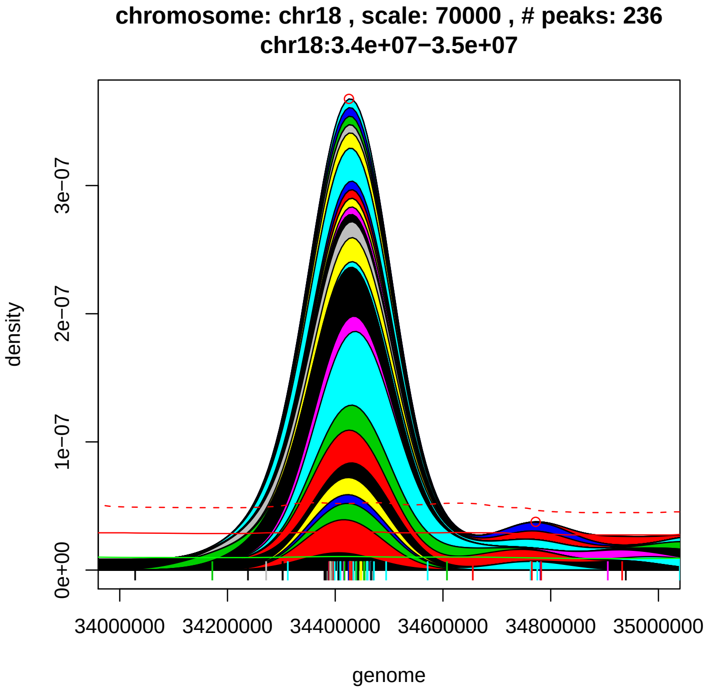

Figure 3: The ‘lava’ plot.

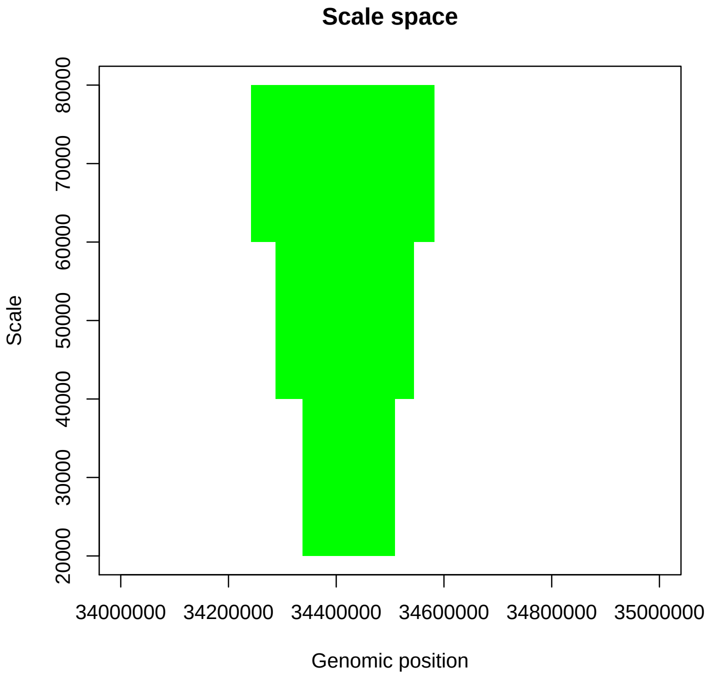

Figure 4: The scale space plot.

3.3 Scale space

To inspect the scale space one can make a scale space plot with the following command:

```
> plot(ca, type = "scale.space", chr = "chr18", bpLim = c(3.4e+07,
+     3.5e+07), interactive = FALSE)
```

The result is depicted is Figure 4. The green blocks represent significant regions, obtained by computing the region corresponding to a peak exceeding the significance threshold. In this case, the CIS becomes smaller as the scale decreases.

4 Export

Several functions are included to export the results of a cimpl analysis. First, one can retrieve the CISs in a `data.frame` using the `getCISs` function. This function also associates genes to CISs by associating the closest gene to the CIS.

```
> genes <- getEnsemblGenes(ca)
> ciss <- getCISs(ca, scales = 50000, genes = genes)
> ciss[1:3, ]
```

```
                   chromosome peak_location peak_height    start      end
CIS18:34427530_50k      chr18      34427530   50.302924 34282128 34543852
CIS18:7901316_50k       chr18       7901316   11.967687  7838309  7954631
CIS18:50077657_50k      chr18      50077657    7.542657 50053423 50106737
                    width n_insertions p_value scale associated_ensembl_gene_id
CIS18:34427530_50k 261725           64       0 50000         ENSMUSG00000005871
CIS18:7901316_50k  116323           13       0 50000         ENSMUSG00000024283
CIS18:50077657_50k  53315            9       0 50000         ENSMUSG00000037416

CIS18:34427530_50k ENSMUSG00000073607|ENSMUSG00000014504|ENSMUSG00000005873|ENSMUSG0000001
CIS18:7901316_50k
CIS18:50077657_50k
                   associated_external_gene_id
CIS18:34427530_50k                         Apc
CIS18:7901316_50k                          Wac
CIS18:50077657_50k                       Dmxl1
                                           other_external_gene_id
CIS18:34427530_50k AC114003.4-1|Srp19|Reep5|Pkd2l2|SNORA17|Fam13b
CIS18:7901316_50k                                 AC130718.3-1|U6
CIS18:50077657_50k                                    AC120859.12
```

To inspect the full results of a genome-wide, multi-scale analysis, we have included the `export.html` function.

```
> export.html(ca, genes = genes, dir = "colorectal_chr18", verbose = FALSE)
```

Open `[CIMPL package dir]/doc/colorectal_chr18/index.html` with your browser.

Additionally, export functions for BED and WIG files (`export.bed` and `export.wig`) are also available.

References

[1] De Ridder *et al.* Detecting statistically significant common insertion sites in retroviral insertional mutagenesis screens. *PLoS Comput Biol*, 2:e166, 2006.

[2] Starr *et al.* A transposon-based genetic screen in mice identifies genes altered in colorectal cancer. *Science.* 2009 Mar 27;323(5922):1747-50. Epub 2009 Feb 26.

Session info

```
> sessionInfo()

R version 2.10.0 alpha (2009-09-30 r49906)
i386-apple-darwin9.8.0

locale:
[1] C
```

```
attached base packages:
[1] tools     stats     graphics  grDevices utils     datasets  methods  
[8] base     

other attached packages:
[1] BSgenome.Mmusculus.UCSC.mm9_1.3.11 BSgenome_1.13.14                  
[3] cimpl_0.99.21                      xtable_1.5-5                      
[5] biomaRt_2.1.0                      MASS_7.3-2                        
[7] KernSmooth_2.23-3                  Biostrings_2.13.47                
[9] IRanges_1.3.86                    

loaded via a namespace (and not attached):
[1] Biobase_2.5.8 RCurl_0.94-1  XML_2.6-0    
```