peakC tutorial

Geert Geeven & Elzo de Wit

April 4, 2018

It is becoming increasingly clear that the 3D organization of the genome plays an important in the regulation of genes. To measure the 3D conformation of the genome various 3C-based have been developed (4C/5C/Hi-C/Capture-C/Promoter-Capture-Hi-C). 4C, Capture-C and PCHiC measure the contact frequency of a single region ("viewpoint" or "bait") with the rest of the genome.

The main goal of these experiments is to identify regions (which we will call "peaks") whose contact frequencies with the viewpoint region are enriched over the background. Because of the non-uniform distribution of background contact frequencies this is not a trivial exercise. In order to model the background, we assume that the contact frequency only decreases with an increase in distance from the viewpoint (i.e. monotonically decreases). peakC achieves this by performing monotonic regression using the `isotone` package. For a more detailed discussion of the theoretical background and performance of the package we refer the reader to our paper: (reference pending). The current tutorial focuses on how to perform 4C/Capture-C/PCHiC data analysis using peakC.

In order to load the package into the R environment run the following command:

```
library(peakC)
```

Data structure

The basic data structure for a (single) 4C/Capture-C/PCHiC experiment is a two-column matrix, with one column for the position of the fragment (end) and one column with the number of reads (coverage, 4C signal). We can use one of the peakC functions to read in the 4C/Capture-C/PCHiC data. As an example we use the `readqWig` function, which can read in *variableStep* wiggle files, which are compatible with the UCSC genome browser. The data we read in is a 4C experiment for one of the hemoglobin genes (alpha-globin) in fetal livers of mice.

```
data <- readqWig("data/alpha-globin_FL_1_cis.wig.gz", vp.pos=32224333, window=700e3)
head(data$data)
```

```
##      frag_pos frag_score
## [1,] 31526039   0.000000
## [2,] 31526481 121.623265
## [3,] 31527169   4.504565
## [4,] 31528018   0.000000
## [5,] 31529060  90.091308
## [6,] 31529432   0.000000
```

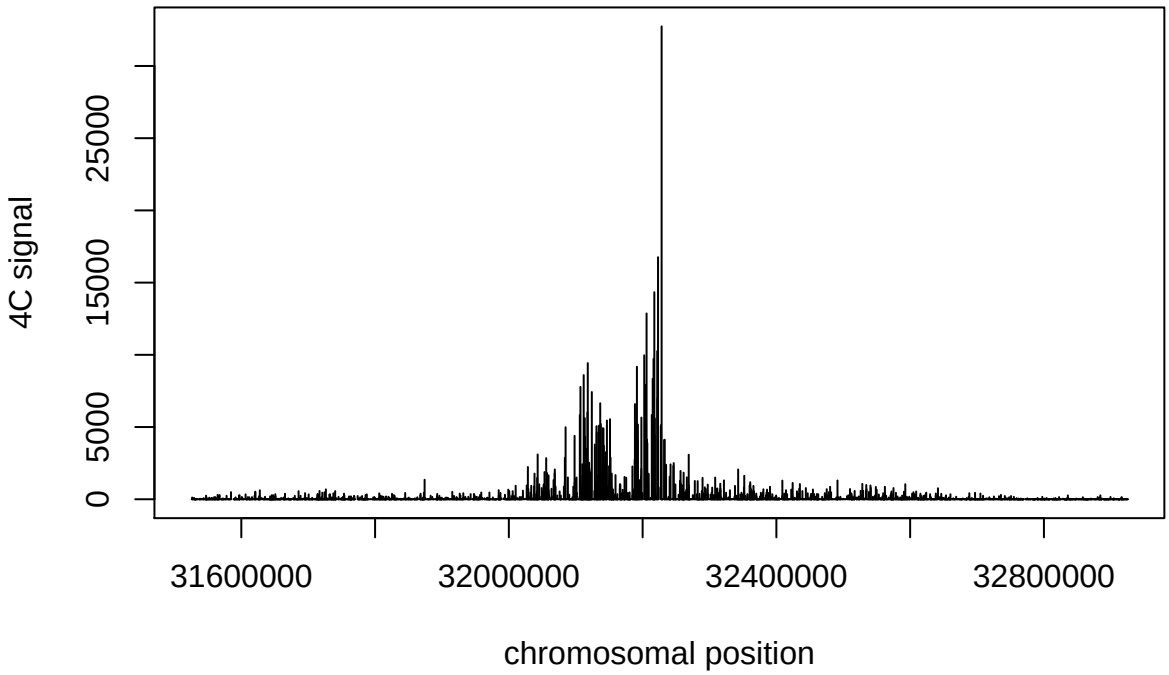
The data structure contains an entry for the 4C data `data$data` and the quality metrics of the experiment (`data$quality`). The quality metrics will be discussed at the end of the tutorial. The `readqWig` function has selected the fragments in the 700kb genomic region downstream and upstream of the viewpoint (position set by `vp.pos`). The flanking window size can be set with `window`. The coverage score is automatically normalized to 1 million intrachromosomal reads (reads in cis).

Nota bene As you can see in the data structure the 1st, 4th and 6th row have zeroes in the data. For peakC to function properly it is necessary to leave the zeroes (i.e. uncovered regions) in the dataset. A lack of captured fragments is also information that is required for the proper calculation of the background model (see below).

Alternative functions for reading in data are:

- `readMatrix`: reads a simple two column file (wig file without the header)
- `readMultiple`: reads multiple matrix files at once, returns a list
- `readMultipleWig`: like read multiple, but for wig files
- `readMultiColumnFile`: reads a multiple column file, 1st column is fragment position, the rest of the columns are 4C/Capture-C/PCHiC data columns

A plot of the raw 4C data looks like this:



The expression of the alpha-globin gene is extremely high in this tissue and is regulated by enhancers that flank this gene. We can already see the promoter-enhancer contacts appearing in the raw data. But peakC can be used to detect the "peak" regions that contact the viewpoint regions significantly above the background frequency.

Identifying contacted regions ("peaks")

4C analysis with replicate experiments

In order to identify peaks we *strongly* advise using replicates. For this tutorial we have analyze data from a 4C experiment of the alpha-globin gene which was done in triplicate. The peak calling consists of two steps. 1) Reading the data (shown above). 2) Statistical analysis using the `combined.analysis` function. A third optional step is to plot the 4C data and the peaks that were identified.

Below we show the simplest way to run the peakC peak caller. Note that as default it runs with a 700kb flanking region (`window`), a running window size of 21 and an FDR threshold of 0.1. For details on the required and optional parameters see the man page. We refer the reader to our paper for guidance on how to set appropriate significance thresholds.

```
viewpoint <- 32224333
files <- dir(path="data/", pattern="alpha-globin_FL_[123]_cis.wig.gz", full=T)
data <- readMultipleWig(files, vp.pos = viewpoint)
res <- combined.analysis(data, vp.pos = viewpoint)
```

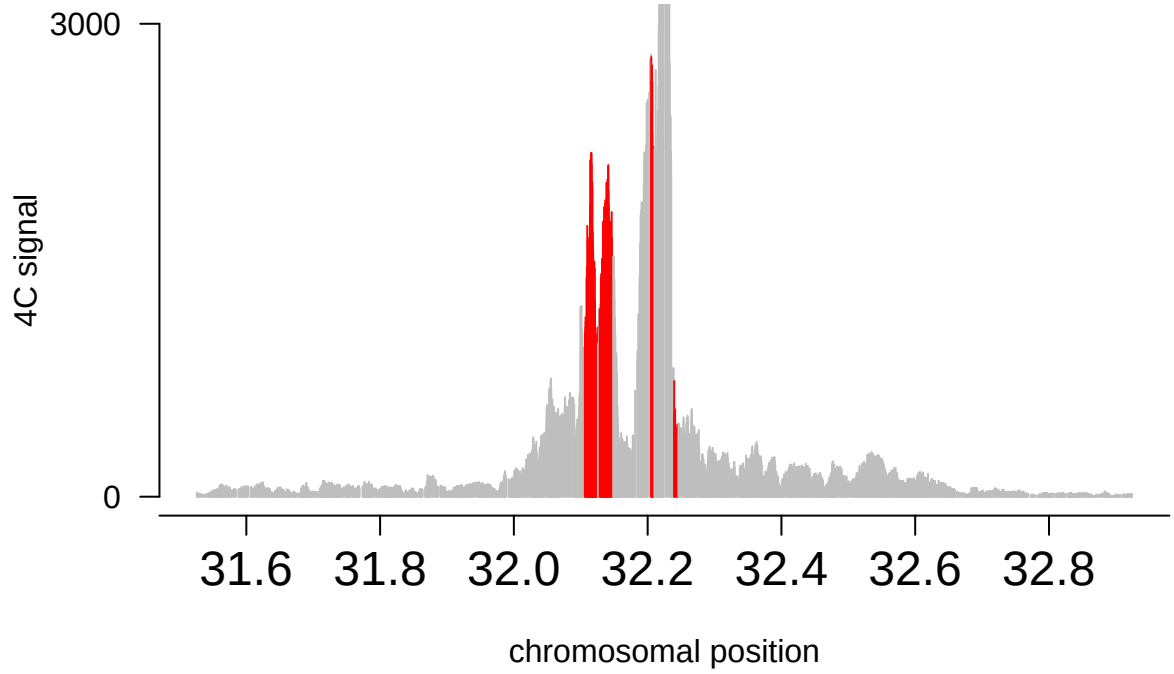
```
## Loading required package: isotone
```

```
head(res$peak)
```

```
## [1] 32106117 32107017 32107166 32107433 32107829 32108173
```

`combined.analysis` outputs a list of with the results of the peak calling (stored here in `res`). The contents of the results list will be discussed in detail below. What is important for now is the `peak` vector in the list contains the fragments that represent significant interactions with the viewpoint. We can use the output of `combined.analysis` to create a plot by calling the `plot_C` function. The significant fragment windows are shown in red.

```
plot_C(res)
```



Single experiment 4C analysis

Below an explanation will be given for performing single experiment analysis. Please note that in a single experiment analysis no formal statistical analysis can be performed and the selection of peak fragments relies on thresholds for the differences and ratios of fragments with respect to the background model fitted 4C coverage. In general it is hard to control for spurious peaks which are not reproducible between 4C experiments. This is discussed in more detail in the companion paper.

First read in the data:

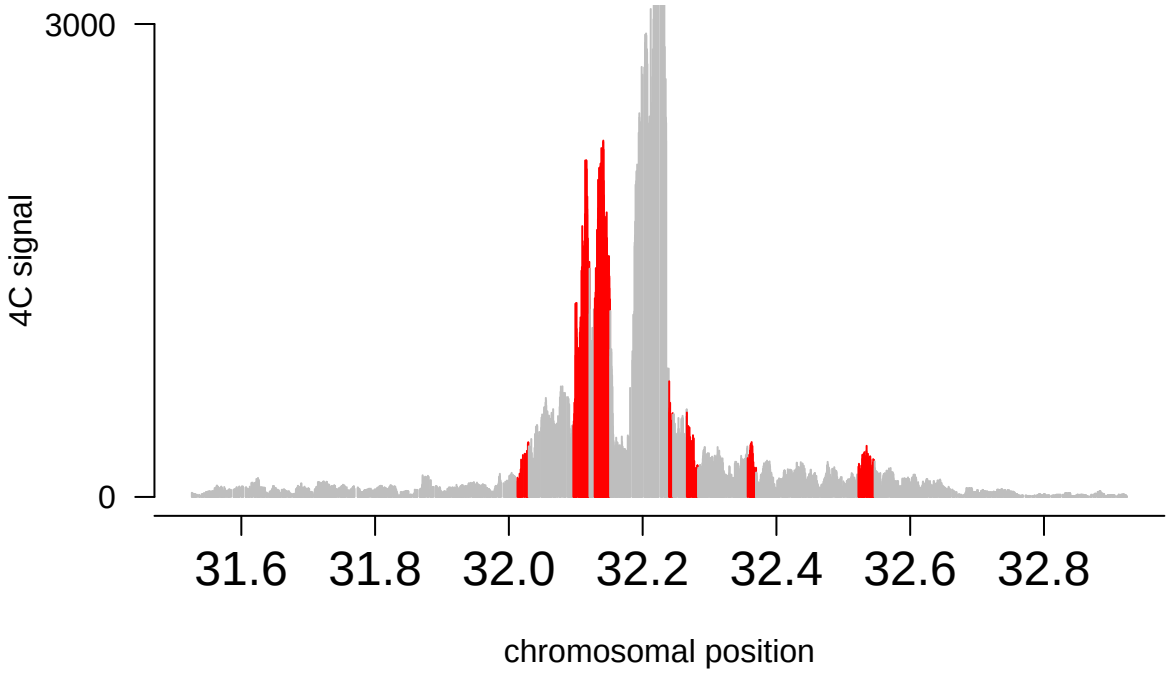
```
data <- readqWig("data/alpha-globin_FL_1_cis.wig.gz", vp.pos=32224333, window=700e3)
```

and the run the command for identifying the peaks

```
res <- single.analysis(data$data, vp.pos=32224333, qWd = 2.5)
```

TBoth qWd (difference with respect to background) and qWr (ratio) can be varied to increase or lower the stringency of the peak calling. To plot the significant fragments, we again use the `plot_C` function.

```
plot_C(res)
```



The single experiment peak caller calls most of the peaks that were called by the single experiment peak caller. However, an additional set of peaks are called, some of which are likely false positive peaks.

Multi-column Capture-C experiment

In addition to 4C experiments peakC can also analyse Capture-C data. Below an example is shown how to identify significant fragments from a Capture-C experiment with 4 replicates. This data was taken from (Davies et al. 2015). A file with multiple columns can be read as follows and return a list of data.frames.

```
data <- readMultiColumnFile("data/Hba_cis.txt", vp.pos = 32189804, window=300e3)
head(data[[1]])
```

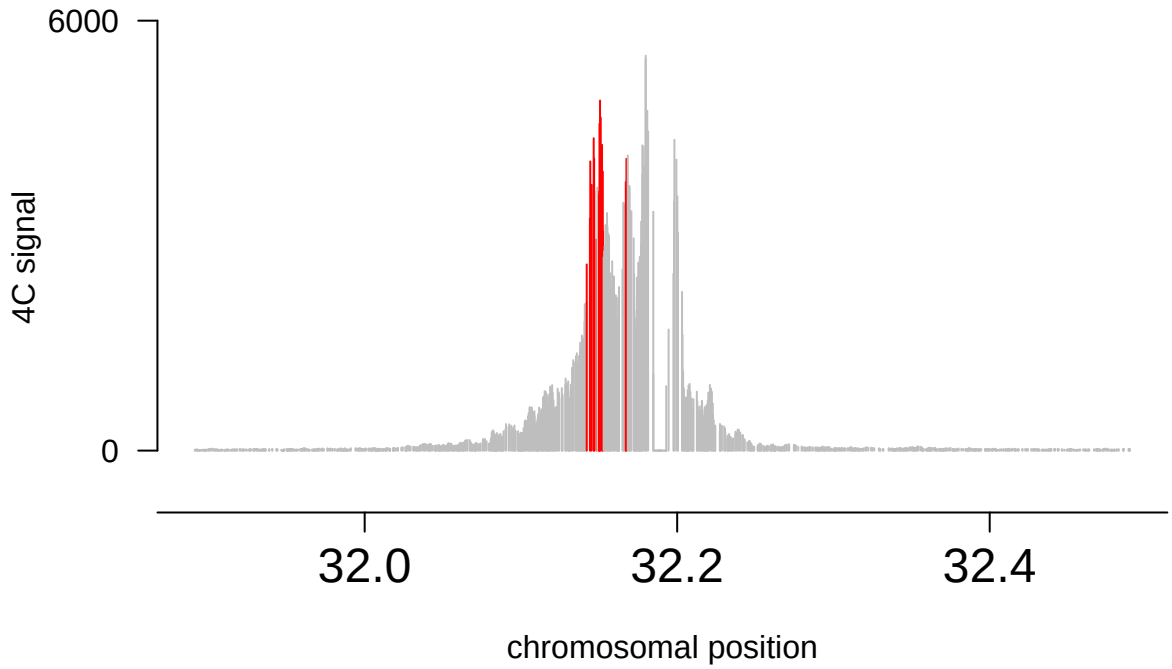
```
##       frag_pos frag_score
## 23826 31891309  32.730928
## 23827 31891573  28.055081
## 23828 31892305   0.000000
## 23829 31892486  18.703388
## 23830 31893036  23.379235
## 23831 31893231   4.675847
```

```
head(data[[2]])
```

```
##       frag_pos frag_score
## 23826 31891309   0.000000
## 23827 31891573   0.000000
## 23828 31892305   5.412749
## 23829 31892486  10.825498
## 23830 31893036  43.301994
## 23831 31893231  10.825498
```

The list can be passed to the `combined.analysis` function as follows:

```
vp.pos <- c(32182969,32196638)
res <- combined.analysis(data=data, num.exp = 2, vp.pos = vp.pos, wSize = 5)
plot_C(res, y.max=6000, y.min=-600)
```



Note that in the these Capture-C experiments two baits were used. We would like to exclude the sequence in between the baits, which is why we use two values for the position of the viewpoint. The fragments between the baits are excluded from the

Due to the higher resolution of the Capture-C experiment we have changed the window size, i.e. the number fragments over which the averaging is performed, `wSize` to 5. Even though we read in 4 experiments, when we set `num.exp` to 2 only the first two experiments are used for the peak calling.

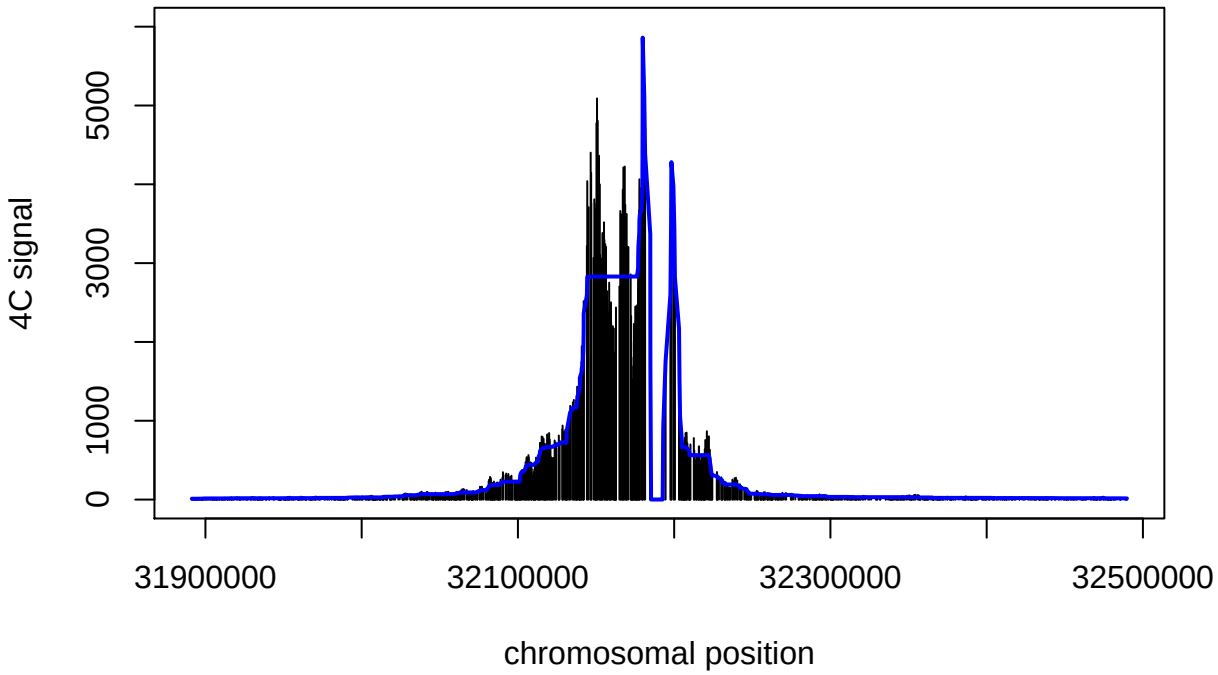
In the function `plot_C` we can play with the limits of the vertical axis by varying `y.min` and `y.max`. By setting the y.min to a negative value it is possible to insert annotations such as gene tracks or ChIPseq data.

Note that there is a difference between the 4C and the CapC data because there is a 43kb difference in the position of the bait or viewpoint.

Output details

Let's look into the resulting data structure in bit more detail. For the two replicate Capture-C experiment the result is a list with 6 elements. Let's go over them: `res$dbR` contains the original data (1st column fragment position, columns 2 until n+1 contain the n experiments). Columns n+2 until 2n+1 contain the estimated background model.

```
plot(res$dbR[,1], res$dbR[,2], type='h', ylim=c(0,6000), xlab="chromosomal position", ylab="4C signal")
lines(res$dbR[,1], res$dbR[,4], lwd=2, col='blue')
```



First we plot the raw data in the second column, next we add the background model shown by the blue line.

`res$peak` contains the fragments that have been called as significant peaks.

`res$num.exp` stores the number of experiments that are stored in `res`.

`res$p.value` contains for every fragment a rank product probability score whether the fragment is a peak fragment.

`res$sel` contains the fragments that are used in the analysis; fragments too close to the viewpoint are not used in the analysis. This distance can be set using the `minDist` parameter in `combined.analysis`.

Quality control

To determine the similarity between experiments we have written a function that performs a pairwise similarity analysis. We use the Jaccard index as a similarity metric. For the Jaccard index the overlap or intersection of the peak fragments is divided by the union of the fragments. Please read our companion paper for a more detailed analysis.

First we will read in a dataset with six replicates of the Oacyl viewpoint in mouse embryonic stem cells.

```
inFiles <- paste0("data/peakC_tutorial_Oacyl_6_reps_mappedReads.txt/set2_viewpoint_1_replicate_",
                  1:6,".txt.gz")

# the genomic position of the Oacyl viewpoint
vp.pos <- 65923816

# read from 2 column matrix files
data <- readMultiple(files=inFiles,vp.pos=vp.pos,window=1.2e6)
```

After reading in the 6 replicates in the data structure we can take a look at the quality characteristics of the dataset. peakC report a number of quality metrics: the percentages of captured/covered fragments in the 100kb and 1Mb flanking the viewpoint and of the total covered fragments in cis (i.e. the whole chromosome of the viewpoint). Also the total number of reads in cis is also reported. Note that reads on non-viewpoint

chromosomes are left out of this analysis. For the dataset that we have just read in the quality metrics are the following:

```
data$quality
```

```
## $percentage.capture.100kb
## [1] 93.76694 93.22493 92.68293 94.03794 94.03794 93.49593
##
## $percentage.capture.1Mb
## [1] 61.46764 60.92608 59.78879 61.14270 60.95315 62.11752
##
## $percentage.capture.cis
## [1] 2.846714 2.778274 2.740664 2.773342 2.748062 2.902205
##
## $total.read.cis
## [1] 135388 165710 102728 141234 158049 155946
```

To compare the pairwise repoducibility we run the dataset through the `pairwise.jaccard` function. Note that this function has the same input parameters as the `single.analysis` function. See below what precise parameters we have chosen for the current analysis.

```
# set threshold parameters for peakC in single experiments
wSize <- 21; qWd <- 2.5; qWr <- 1.0; minDist <- 15e3
jac.mat <- pairwise.jaccard(data = data, vp.pos = vp.pos, wSize = wSize, qWd = qWd, qWr = qWr, minDist =
```

The result is a symmetrical matrix `jac.mat`, in which the Jaccard indices for all pairwise comparisons are stored.

```
jac.mat
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.0000000 0.7009967 0.5958084 0.6588235 0.5436893 0.5518395
## [2,] 0.7009967 1.0000000 0.4498645 0.5799257 0.5352564 0.5481728
## [3,] 0.5958084 0.4498645 1.0000000 0.5432526 0.5873016 0.6072607
## [4,] 0.6588235 0.5799257 0.5432526 1.0000000 0.6386555 0.6042553
## [5,] 0.5436893 0.5352564 0.5873016 0.6386555 1.0000000 0.6387833
## [6,] 0.5518395 0.5481728 0.6072607 0.6042553 0.6387833 1.0000000
```

Note that by changing the parameters `qWd`, `qWr` and `wSize` it is possible to increase or decrease the Jaccard similarity. Making the analysis more or less stringent.

References

Davies, James O J, Jelena M Telenius, Simon J McGowan, Nigel A Roberts, Stephen Taylor, Douglas R Higgs, and Jim R Hughes. 2015. "Multiplexed analysis of chromosome conformation at vastly improved sensitivity." *Nature Methods* 13 (1). Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved.: 74–80. doi:10.1038/nmeth.3664.