BayesPeak

October 5, 2010

`bayespeak` *BayesPeak - Bayesian analysis of ChIP-seq data*

Description

BayesPeak - Bayesian analysis of ChIP-seq data. This function divides the genome into jobs, and performs the BayesPeak algorithm on each using a C backend. The jobs can be performed in parallel, using the package `multicore`. Results are returned in R.

Usage

```
bayespeak(treatment, control, chr = NULL, start,
end, bin.size = 100L, iterations = 10000L,
repeat.offset = TRUE, into.jobs = TRUE, job.size = 6E6L,
job.overlap = 20L, use.multicore = FALSE,
mc.cores = getOption("cores"),
prior = c(5, 5, 10, 5, 25, 4, 0.5, 5))
```

Arguments

`treatment, control`

These arguments should contain the treated ChIP-seq data and the control data, respectively.

Each of these arguments can be:

- a path to a .bed file (this file will be read in as per `read.bed`).
- OR a `data.frame`, which should have columns `"chr", "start", "end", "strand"`.
- OR a `RangedData` object. This object is expected to be split into spaces by chromosome, and should have a data track labelled "strand".

The `control` argument is entirely optional. (Mathematically, leaving this argument out is equivalent to setting gamma = 1 in the model.)

Strand information is expected to be given as "+" or "-".

`chr`

Character vector, specifying which chromosomes to restrict analysis to. Chromosome names must be specified exactly as they appear in the treatment and control arguments.

If left as the default value `chr = NULL`, then BayesPeak will find all chromosomes present in the `treatment` file.

`start, end` Numeric. Locations on the chromosome to start and end at, respectively. If unspecified, then the algorithm will start and end at the minimum and maximum reads found in the data, respectively.

`bin.size` Numeric. Reads are collected into bins. This parameter controls the width of each bin.

`iterations` Numeric. Number of iterations to run the Monte Carlo analysis for.

`repeat.offset` Logical. If `TRUE`, the algorithm is run a second time, this time with the bins offset by `floor(window/2)`.

`into.jobs` Logical. By default, BayesPeak will divide a large region into smaller jobs and analyse each one separately. To prevent this behaviour, set `into.chunks = FALSE`. This may put BayesPeak at increased risk of overflow and underflow issues, and will additionally prevent usage of the parallel processing options.

`job.size` Numeric. The size of the jobs in base pairs, as described above.

`job.overlap` Numeric. Jobs are expanded to overlap each other. This is prevent peaks on the boundary between two jobs being missed. `job.overlap` corresponds to the number of bins by which each job is expanded.

`use.multicore` Logical. If `use.multicore = TRUE`, then the individual chunks will be processed in parallel, using the `multicore` package. The `multicore` package must be installed for this feature to be enabled. At time of writing, it can be downloaded from http://www.rforge.net/multicore/index.html.

`mc.cores` Numeric. The number of cores to be used for parallel processing. This argument is passed directly to the `mclapply` function.

`prior` Numeric. A vector, specifying the prior on the hyperparameters as follows. We have lambda_0 ~ gamma(alpha_0, beta_0) and lambda_1 ~ gamma(alpha_1, beta_1). Additionally, we have that alpha_0, alpha_1, beta_0, beta_1 all have gamma priors. This argument should be c(alpha_0 shape, alpha_0 scale, beta_0 shape, beta_1 scale, alpha_1 shape, alpha_1 scale, beta_1 shape, beta_1 scale).

Details

BayesPeak uses a fully Bayesian hidden Markov model to detect enriched locations in the genome. The structure accommodates the natural features of the Solexa/Illumina sequencing data and allows for overdispersion in the abundance of reads in different regions. Markov chain Monte Carlo algorithms are applied to estimate the posterior distributions of the model parameters, and posterior probabilities are provided for the sites of interest.

Value

A list of 3 objects:

- peaks: A `data.frame` corresponding to the bins that BayesPeak has identified as potentially being enriched. `chr, start, end` give the genomic co-ordinates of the bin. `PP` refers to the posterior probability of the bin being enriched. `job` is the number of the job within which the bin was called, which corresponds to a row in the QC data.frame (see below).

- QC: details of each individual job, listed in columns as follows:

  - `calls` is the number of potentially enriched bins identified in a job (i.e. bins with `PP` > 0.01).

- `score` is simply the proportion of potentially enriched bins with a PP value above 0.5. Intuitively, a larger score is "better", as it indicates that more of the PP values have tended to 0 or 1.
- `chr, start, end` are the genomic co-ordinates of the job.
- We report the average value, across iterations of the algorithm, of the important parameters `p, theta, lambda0, lambda1, gamma` and the average log likelihood `loglhood`.
- `var` is the variance of the bin counts.
- `autocorr` is an estimate of the first order autocorrelation of bin counts.
- `status` indicates whether the job was normal, or offset by half a bin width.

- call: the line of code used to run BayesPeak.

Note that the raw output of this function is not intended to be used directly as results - the output should be summarised using the `summarise.peaks` function before using it in later analysis.

Author(s)

Christiana Spyrou and Jonathan Cairns

References

Spyrou C, Stark R, Lynch AG, Tavare S BayesPeak: Bayesian analysis of ChIP-seq data, BMC Bioinformatics 2009, 10:299 doi:10.1186/1471-2105-10-299

See Also

`read.bed`, `summarise.peaks`.

Examples

```
dir <- system.file("extdata", package="BayesPeak")
treatment <- file.path(dir, "H3K4me3reduced.bed")
input <- file.path(dir, "Inputreduced.bed")

##look at specific region 92-95Mb on chromosome 16
##(we've used half the number of iterations here to reduce the time this example takes)
raw.output <- bayespeak(treatment, input, chr = "chr16", start = 9.2E7, end = 9.5E7, iter
output <- summarise.peaks(raw.output)
output

## Not run:
##analyse all data in file
raw.output.wg <- bayespeak(treatment, input, use.multicore = TRUE)
output <- summarise.peaks(raw.output.wg)

## End(Not run)
```

raw.output — *Example raw.output object*

Description

This data set is an example of the output obtained from the bayespeak() function, in particular as an example of job parameters. The ChIP-seq experiment in question investigates ER binding in cells from the MCF7 cell line.

To keep the size of BayesPeak down, raw.output$peaks has been truncated - only the peaks on chromosome 16 are given. raw.output$QC has not been truncated in any way.

Usage

```
raw.output
```

Format

A list of 3 objects. See `bayespeak` for more details.

References

Many thanks to Dr. Jason Carroll's group for permission to use this data set.

read.bed — *BayesPeak - Bayesian analysis of ChIP-seq data*

Description

Read a .bed file into a data frame, but only the chr, start, end and strand columns.

Usage

```
read.bed(filename, chr)
```

Arguments

`filename`	Character - The path to the .bed file in question.
`chr`	Character vector, specifying which chromosomes to read in. Chromosome names must be specified exactly as they appear in the .bed files. If `chr` is missing, then `read.bed` will read in the entire data set.

Details

The purpose of this function is to extract 4 columns from a bed file: chromosome, start, end and strand. These are assumed to be in columns 1, 2, 3 and 6 respectively.

If the first line begins with "track" then it will be skipped.

The strand sense is expected to be given as "+"/"-".

Value

A RangedData object, split into spaces by chromosome. This object has a "strand" data track.

See the IRanges package vignette for more information.

Author(s)

Jonathan Cairns

References

UCSC BED format FAQ - http://genome.ucsc.edu/FAQ/FAQformat.html#format1

See Also

bayespeak.

Examples

```
dir <- system.file("extdata", package="BayesPeak")
file <- file.path(dir, "H3K4me3reduced.bed")

treatment <- read.bed(file)
treatment
```

---

summarise.peaks *Summarise Peaks*

Description

Summarise Peaks - Combine the potentially enriched bins found by BayesPeak into contiguous peaks, and associate each with a posterior probability.

Usage

```
summarise.peaks(x, threshold = 0.5, method = c("lowerbound", "max"), exclude.job
```

Arguments

x	Raw output from the function bayespeak.
threshold	Numeric vector. threshold must have length equal to either 1 or nrow(x$QC) (i.e. the number of jobs). • If threshold is of length 1, then for each job, all bins with a posterior probability (PP) lower than threshold will be discarded before summarising. • If threshold is of length nrow(x$QC), then jobs are taken to have separate thresholds - in other words, bins in job i will be discarded if they have a PP less than threshold[i]. Note that this behaviour is irrespective of how many jobs are excluded (see the exclude.jobs argument below) - excluded jobs are still assigned a PP threshold, which is essentially ignored.
method	The method used to combine the posterior probabilities of multiple peaks. Current methods are:

- `lowerbound`: A lower bound is found for the posterior probability of the region containing a peak. In contiguous regions of moderately high probability, this method should report a fairer value than `method = max`. Suppose we have a set of n non-intersecting calls within our region, with posterior probabilities p_1 to p_n respectively of containing peaks. Then the probability of there being a peak in this region is at least $1 - \prod_{i=1}^{n}(1 - p_i)$. We maximise this over all possible sets of non-intersecting calls within the region. (Usually, this will simply be a choice between exclusively using the offset or the non-offset analyses.)
- `max`: Combined region has probability equal to the maximum posterior probability over all the peaks it contains.

`exclude.jobs` A vector of integers, denoting jobs to be excluded from later analysis. Alternatively, a logical vector (to be passed through `which()`).

Value

A RangedData object corresponding to the peaks called - each range has an associated PP (Posterior Probability) value.

Author(s)

Jonathan Cairns

See Also

bayespeak.

Examples

```
dir <- system.file("extdata", package="BayesPeak")
treatment <- file.path(dir, "H3K4me3reduced.bed")
input <- file.path(dir, "Inputreduced.bed")

##look at specific region 92-95Mb on chromosome 16
##(we've used half the number of iterations here to reduce the time this example takes)
raw.output <- bayespeak(treatment, input, chr = "chr16", start = 9.2E7, end = 9.5E7, iter
output <- summarise.peaks(raw.output)
output

##higher threshold
output.ht <- summarise.peaks(raw.output, threshold = 0.9)
output.ht
```

Index