# Package ‘epiG’

June 12, 2016

**Type** Package

**Title** Statistical Inference of Epi-allelic Patterns from Whole-Genome Bisulphite Sequencing Data

**Version** 0.9.0

**Date** 2016-06-12

**Author** Martin Vincent

**Maintainer** Martin Vincent <martin.vincent.dk@gmail.com>

**Description** Statistical method to infer epi-
allelic haplotypes, annotated with CpG methylation marks and polymorphisms, from whole-genome bisulphite sequencing data, and nucleosome occupancy from NOMe-seq data

**URL**

**License** GPL (>= 2)

**LazyLoad** yes

**Depends** R (>= 3.0.0)

**LinkingTo** Rcpp, RcppProgress, RcppArmadillo

**RoxygenNote** 5.0.1

**NeedsCompilation** yes

## R topics documented:

## `auto_config` *Create Standard Configuration*

### Description

Create a epiG configuration with standard parameters

### Usage

```
auto_config(ref_file, alt_file, bam_file, chr, start, end, seq_type = "BSeq",
  paired_reads = NULL, min_CG = NULL, min_HCGD = NULL, min_DGCH = NULL,
  min_overlap = NULL, ...)
```

## Arguments

| | |
|---|---|
| `ref_file` | genome reference file (path to .fa file) |
| `alt_file` | alternative nucleotide file (path to .fa file) |
| `bam_file` | bam file (path to .bam file) |
| `chr` | reference name |
| `start` | start position of region to processes |
| `end` | end position of region to processes |
| `seq_type` | sequencing type ("BSeq" for bisulphite sequencing, "NOMeSeq" for NOMe sequencing) |
| `paired_reads` | should pair information be used (TRUE/FALSE or NULL, if NULL then paired information is used if pairs are present in bam file) |
| `min_CG` | minimum overlapping CG positions (if NULL deafult value is used) |
| `min_HCGD` | minimum overlapping HCGD positions (if NULL deafult value is used) |
| `min_DGCH` | minimum overlapping DGCH positions (if NULL deafult value is used) |
| `min_overlap` | minimum overlapping length (if NULL deafult value is used) |
| `...` | additional arguments (overrides default values) |

## Value

An epiG configuration

## Author(s)

Martin Vincent

## Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")
ref_file <- system.file("extdata", "hg19_GNAS.fa", package="epiG")
alt_file <- system.file("extdata", "dbsnp_135.hg19_GNAS.fa", package="epiG")

# Specify region
chr <- "chr20"
start <- 57400000
end <- 57400000 + 1000

# Build epiG configuration
config <- auto_config(
bam_file = bam_file,
ref_file = ref_file,
alt_file = alt_file,
chr = chr,
start = start,
end = end,
# If ref_file and alt_file contains the entire chromosome this is not needed
ref_offset = 57380000,
alt_offset = 57380000)

config # print a summary of the configuration

# Run epiG
fit <- epiG(max_threads = 2, config = config)

fit # print a summary of inferred model
```

## `BSeq` *Bisulphite Conversion Model*

### Description

Create a bisulphite sequencing conversion model

### Usage

```
BSeq(bisulphite_rate = 0.95, bisulphite_inap_rate = 0.05, Lmax = 110, ...)
```

### Arguments

`bisulphite_rate`
: bisulphite conversion rate (numeric in the range (0, 1])

`bisulphite_inap_rate`
: bisulphite inappropriate conversion rate (numeric in the range (0, 1])

`Lmax` maximal read length (integer)

`...` ignored

### Value

an epiG conversion model

### Author(s)

Martin Vincent

### Examples

```
BSeq()
```

## `chain_info` *Information about chains*

### Description

Rerive information about infered haplotype chains

### Usage

```
chain_info(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

A data.frame with `nchain(object)` rows. With the following columns:

| | |
|---|---|
| `chain.id` | a unique chain id |
| `start` | first position of the chain |
| `end` | last position of the chain |
| `length` | length of the chain |
| `nreads` | number of reads in the chain |
| `nreads.fwd` | number of fwd reads in the chain |
| `nreads.rev` | number of rev reads in the chain |
| `depth.fraction` | the computed depth fraction |

### Author(s)

Martin Vincent

### Examples

```
data(example)

chains <- chain_info(fit)

subset(chains, nreads > 2)
```

## end.epiG *End position*

### Description

Return the last position in the model

### Usage

```
## S3 method for class 'epiG'
end(x, ...)
```

### Arguments

| | |
|---|---|
| x | an epiG model |
| ... | ignored |

### Value

the last position in model

### Author(s)

Martin Vincent

### Examples

```
data(example)

end(fit)
```

## epiG *Fit an epiG epigenotype model*

### Description

Fit an epiG epigenotype model

### Usage

```
epiG(config, max_threads = 2L)
```

### Arguments

| | |
|---|---|
| config | epiG configuration |
| max_threads | maximal number of threads to use |

## Value

fitted model

## Author(s)

Martin Vincent

## Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")
ref_file <- system.file("extdata", "hg19_GNAS.fa", package="epiG")
alt_file <- system.file("extdata", "dbsnp_135.hg19_GNAS.fa", package="epiG")

# Specify region
chr <- "chr20"
start <- 57400000
end <- 57400000 + 1000

# Build epiG configuration
config <- auto_config(
bam_file = bam_file,
ref_file = ref_file,
alt_file = alt_file,
chr = chr,
start = start,
end = end,
# If ref_file and alt_file contains the entire chromosome this is not needed
ref_offset = 57380000,
alt_offset = 57380000)

# Run epiG
fit <- epiG(max_threads = 2, config = config)

# Fetch additional information
fit <- fetch_reads(fit)
fit <- fetch_ref(fit)
fit <- fetch_alt(fit)

# Information about fitted model
fit

# Information about haplotype chains
chain_info(fit)
```

# epiG_chunks *Fit epiG epigenotype models*

### Description

Fit epiG epigenotype models

Fit an epiG epigenotype model for each configuration in the list configs.

### Usage

```
epiG_chunks(configs, max_threads = 8L)
```

### Arguments

| | |
|---|---|
| `configs` | list of epiG configurations |
| `max_threads` | maximal number of threads to use |

### Value

list of fitted models

### Author(s)

Martin Vincent

---

## `epiG_config` *Create an epiG Configuration*

---

### Description

Create a custom epiG configuration

### Usage

```
epiG_config(ref_file, alt_file, ref_offset = 1, alt_offset = 1, model,
  min_overlap, min_CG, min_HCGD, min_DGCH, ref_prior = 1 - 1e-04,
  structual_prior = 1, quality_threshold = 0.02, margin = 5,
  chunk_method = "reads", chunk_size = 5000, hard_limit = 6000,
  paired_reads, max_iterations = 1e+05, max_stages = 1, verbose = TRUE,
  ...)
```

### Arguments

| | |
|---|---|
| `ref_file` | genome reference file (path to .fa file) |
| `alt_file` | alternative nucleotide file (path to .fa file) |
| `ref_offset` | ref file offset (usually ref_offset = 1) |
| `alt_offset` | alt file offset (usually alt_offset = 1) |
| `model` | conversion model |
| `min_overlap` | minimum overlapping length |

`min_CG` minimum overlapping CG positions

`min_HCGD` minimum overlapping HCGD positions

`min_DGCH` minimum overlapping DGCH positions

`ref_prior` genotype prior parameter

`structual_prior` structural prior scaling

`quality_threshold` discard reads with mean epsilon quality higher than quality_threshold

`margin` cut off margin

`chunk_method` Method used to split region into chunks ('none' only one chunk, 'reads' chunks of approximately chunk_size reads, 'bases' chunks of chunk_size bases)

`chunk_size` chunk size

`hard_limit` maximal number of reads loaded per chunk (reads not loaded will be completely ignored)

`paired_reads` used pair information (reads with the same name in the bam file is paired and will be forced into the same haplotype chain)

`max_iterations` maximal number of iterations

`max_stages` experimental stage optimization (if <= 1 then stage optimization is off)

`verbose` show information while running

`...` ignored

## Value

an epiG configuration

## Author(s)

Martin Vincent

## Examples

```
# Retrieve paths to raw data files
ref_file <- system.file("extdata", "hg19_GNAS.fa", package="epiG")
alt_file <- system.file("extdata", "dbsnp_135.hg19_GNAS.fa", package="epiG")

config <- epiG_config(

ref_file = ref_file,
alt_file = alt_file,
ref_offset = 57380000,
alt_offset = 57380000,

model = BSeq(),
min_overlap = 80,
min_CG = 0,
min_HCGD = 0,
min_DGCH = 0,

ref_prior = 0.999,
structual_prior = 1,
margin = 5,
quality_threshold = 0.020,

 chunk_method = "reads",
chunk_size = 8000,
hard_limit = 10000,

paired_reads = TRUE,

max_iterations = 1e5,

verbose = TRUE
)

config # print a summary of the configuration

# Specify region and bam file
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

chr <- "chr20"
start <- 57400000
end <- 57400000 + 1000

config <- set_run_configuration(config, bam_file, chr, start, end)

# Run epiG
fit <- epiG(max_threads = 2, config = config)

fit # print a summary of inferred model
```

## fetch_alt *Fetch Alternative Nucleotides*

### Description

Load alternative nucleotides and include it in epiG object.

### Usage

```
fetch_alt(object)
```

### Arguments

| | |
|---|---|
| `object` | a epiG model |

### Value

epiG model

### Author(s)

Martin Vincent

## `fetch_reads` *Fetch Reads*

### Description

Reads will be loaded and include in the epiG object

### Usage

```
fetch_reads(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG epigenotype model |

### Value

model with reads included (this may increase the memory use)

### Author(s)

Martin Vincent

## `fetch_read_info` *Information About Reads*

### Description

Fetch information about reads overlapping the specified region

### Usage

```
fetch_read_info(file, refname, start, end)
```

### Arguments

| | |
|---|---|
| `file` | path to bam file |
| `refname` | reference name |
| `start` | start of region |
| `end` | end of region |

### Value

data.frame with information abut reads. Columns:

| | |
|---|---|
| `name` | name of read |
| `start` | start position of read |
| `end` | end position of read |
| `length` | length of read |

### Author(s)

Martin Vincent

### Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

fetch_read_info(bam_file, "chr20", 57400000, 57400000 + 100)
```

## `fetch_ref` *Fetch Reference Genom*

### Description

Load reference genome and include it in epiG object.

### Usage

```
fetch_ref(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG epigenotype model |

### Value

epiG epigenotype model with reference genome included

### Author(s)

Martin Vincent

## `file_info` *Fetch Bam File Information*

### Description

Fetch information about bam file

### Usage

```
file_info(file)
```

### Arguments

| | |
|---|---|
| `file` | path to bam file |

### Value

a data.frame with the following columns:

| | |
|---|---|
| `ref` | refname |
| `length` | length of ref |
| `nreds` | number of reads assigned to refname |
| `mean_read_length` | the mean read length of reads assigned to refname |

### Author(s)

Martin Vincent

### Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

file_info(bam_file)
```

## `fit` *Example fit*

### Description

TODO

## header_info *Fetch Bam Header*

### Description

Load bam file header

### Usage

```
header_info(file)
```

### Arguments

| | |
|---|---|
| file | path to bam file |

### Value

a list of refnames and lengths associated with the file

### Author(s)

Martin Vincent

### Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

header_info(bam_file)
```

## length.epiG *Length of model*

### Description

Length of model in base pairs

### Usage

```
## S3 method for class 'epiG'
length(x, ...)
```

### Arguments

| | |
|---|---|
| x | epiG model |
| ... | ignored |

### Value

Length of model in base pairs

### Author(s)

Martin Vincent

### Examples

```
data(example)

length(fit)
```

## load_reads *Load Reads*

### Description

Load the reads overlapping the specified region

### Usage

```
load_reads(file, refname, start, end, quality_threshold = 1,
  raw_quality_scores = FALSE)
```

### Arguments

| | |
|---|---|
| `file` | path to bam file |
| `refname` | reference name |
| `start` | start of region |
| `end` | end of region |
| `quality_threshold` | quality threshold |
| `raw_quality_scores` | if TRUE raw quality score will be returned |

### Value

a list with the following entries:

| | |
|---|---|
| `reads` | a list of reads (each read represented by a vector of bases) |
| `quality` | a list quality scores |
| `positions` | a vector of the start positions of the reads |
| `lengths` | a vector of the lengths of the reads |
| `names` | a vector of the names of the reads |

### Author(s)

Martin Vincent

### Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

info <- load_reads(bam_file, "chr20", 57400000, 57400000 + 100)

# Bases, qualities, start position, length and name of first read
info$reads[[1]]
info$quality[[1]]
info$position[1]
info$lengths[1]
info$names[1]
```

---

## `locate_C` *Locate C*

---

### Description

Locate C positions

### Usage

```
locate_C(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG object |

### Value

positions of C in object

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_C(fit)
```

## `locate_CG` *Locate CpG*

### Description

Locate CpG positions in the refrence genom

### Usage

```
locate_CG(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

a vector of CpG positions

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_CG(fit)
```

## `locate_DGCH` *Locate DGCH*

### Description

Locate DGCH (isolated GpC) positions in the refrence genom

### Usage

```
locate_DGCH(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

a vector of isolated GpC positions

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_DGCH(fit)
```

---

## locate_GC *Locate GpC*

---

### Description

Locate GpC positions in the refrence genom

### Usage

```
locate_GC(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

a vector of GpC positions

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_GC(fit)
```

## `locate_HCGD` *Locate HCGD*

### Description

locate HCGD (isolated CpG) positions in the refrence genom

### Usage

```
locate_HCGD(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

a vector of isolated CpG positions

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_HCGD(fit)
```

## `locate_mismatch` *Locate Mismatches*

### Description

Locate positions where at least one chain has a genotype not matching with the reference.

### Usage

```
locate_mismatch(object)
```

### Arguments

| | |
|---|---|
| `object` | an epiG model |

### Value

a vector of postions of mismatches

### Author(s)

Martin Vincent

### Examples

```
data(example)

fit <- fetch_ref(fit)
locate_mismatch(fit)
```

## nchain *Number of chains*

### Description

Number of chains in the model

### Usage

```
nchain(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

number of chains in the model

### Author(s)

Martin Vincent

### Examples

```
data(example)

nchain(fit)
```

## `nchunks` *Number of chunks*

### Description

Number of chunks in the epiG object

### Usage

```
nchunks(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

the number of chunks in the epiG object

### Author(s)

Martin Vincent

### Examples

```
data(example)

nchunks(fit)
```

## `NOMeSeq` *NOMe-sequencing Conversion Model*

### Description

Create a NOMe-seq conversion model

### Usage

```
NOMeSeq(bisulphite_rate = 0.95, bisulphite_inap_rate = 0.05, Lmax = 110,
  ...)
```

### Arguments

| | |
|---|---|
| `bisulphite_rate` | bisulphite conversion rate (numeric in the range (0, 1]) |
| `bisulphite_inap_rate` | bisulphite inappropriate conversion rate (numeric in the range (0, 1]) |
| `Lmax` | maximal read length (integer) |
| `...` | ignored |

### Value

an epiG conversion model

### Author(s)

Martin Vincent

### Examples

```
NOMeSeq()
```

## `nread` *Number of reads in model*

### Description

Number of reads in model

### Usage

```
nread(object)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |

### Value

number of reads contined in model

### Author(s)

Martin Vincent

### Examples

```
data(example)

nread(fit)
```

## `position_info` *Position Info*

### Description

Retriev information about the estimated epi-genotype at a given position in the model

### Usage

```
position_info(object, pos)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |
| `pos` | position |

### Value

??

### Author(s)

Martin Vincent

## `print.epiG` *Print Information About an Fitted epiG Model*

### Description

Print Information About an Fitted epiG Model

### Usage

```
## S3 method for class 'epiG'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | epiG model |
| `...` | ignored |

### Author(s)

Martin Vincent

## print.epiG.config *Print Information About an epiG Configuration*

### Description

Print Information About an epiG Configuration

### Usage

```
## S3 method for class 'epiG.config'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | epiG configuration |
| ... | ignored |

### Author(s)

Martin Vincent

## print.epiG.model *Print Information About an epiG Conversion Model*

### Description

Print Information About an epiG Conversion Model

### Usage

```
## S3 method for class 'epiG.model'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | epiG conversion model |
| ... | ignored |

### Author(s)

Martin Vincent

## `print.epiG.reads` *Print Information About an epiG Reads Object*

### Description

Print information about an epiG reads object

### Usage

```
## S3 method for class 'epiG.reads'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | epiG reads object |
| `...` | ignored |

### Author(s)

Martin Vincent

## `read_count` *Count Reads*

### Description

Count reads overlapping the specified region

### Usage

```
read_count(file, refname, start, end)
```

### Arguments

| | |
|---|---|
| `file` | path to bam file |
| `refname` | reference name |
| `start` | start of region |
| `end` | end of region |

### Value

number of reads and total bps in reads overlapping region

### Author(s)

Martin Vincent

### Examples

```
# Retrieve paths to raw data files
bam_file <- system.file("extdata", "GNAS_small.bam", package="epiG")

read_count(bam_file, "chr20", 57400000, 57400000 + 100)
```

## `read_depth` *Read depth*

### Description

Retrive the read depth at a given position in the model

### Usage

```
read_depth(object, pos = NULL)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |
| `pos` | position |

### Value

??

### Author(s)

Martin Vincent

### Examples

```
data(example)

read_depth(fit)
```

## `read_fasta` *Read FASTA File*

### Description

Read a raw FASTA file

### Usage

```
read_fasta(file, refname, start, len, offset = 1)
```

### Arguments

| | |
|---|---|
| `file` | path to fasta file |
| `refname` | reference name |
| `start` | start of region |
| `len` | length of region |
| `offset` | file offset (position of first base in file, usually offset = 1) |

### Value

a vector of length `len` containing the bases

### Author(s)

Martin Vincent

### Examples

```
ref_file <- system.file("extdata", "hg19_GNAS.fa", package="epiG")

# Specify region
chr <- "chr20"
start <- 57400000
end <- 57400000 + 100

#Note that usually offset = 1
read_fasta(ref_file, chr, start, len = end-start+1, offset = 57380000)
```

## `read_info` *Information about reads*

### Description

Retrive information about reads in the model

### Usage

```
read_info(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |
| `...` | additonal arguments |

### Value

??

### Author(s)

Martin Vincent

## `read_info.epiG` *Information about reads*

### Description

Retrive information about reads in the model

### Usage

```
## S3 method for class 'epiG'
read_info(object, inc.symbols = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |
| `inc.symbols` | if `TRUE` each line in the returned data.frame will correspond to one nuleobase, if `FALSE` one read. |
| `...` | ignored |

### Value

??

### Author(s)

Martin Vincent

---

## read_info.epiG_reads

*Information about reads*

---

### Description

Retrive information about reads from a epiG read object produced with `load_reads`

### Usage

```
## S3 method for class 'epiG_reads'
read_info(object, inc.symbols = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `object` | epiG read object |
| `inc.symbols` | if `TRUE` each line in the returned data.frame will correspond to one nuleobase, if `FALSE` one read. |
| `...` | ignored |

### Value

??

### Author(s)

Martin Vincent

---

## set_run_configuration

*Run Configuration*

---

### Description

set run configuration

### Usage

```
set_run_configuration(config, filename, refname, start, end)
```

### Arguments

| | |
|---|---|
| config | an epiG configuration |
| filename | bam file (path to .bam file) |
| refname | reference name |
| start | start position of region to processes |
| end | end position of region to processes |

### Value

an epiG configuration

### Author(s)

Martin Vincent

### Examples

```
# See epiG_config example
```

## start.epiG *Start position*

### Description

Return the first position in the model

### Usage

```
## S3 method for class 'epiG'
start(x, ...)
```

### Arguments

| | |
|---|---|
| x | an epiG model |
| ... | ignored |

### Value

the first position in the model

### Author(s)

Martin Vincent

### Examples

```
data(example)

start(fit)
```

## `subregion` *Subregion*

### Description

Exatract a subregion of an epiG model

### Usage

```
subregion(object, start, end, chop.reads = FALSE)
```

### Arguments

| | |
|---|---|
| `object` | epiG model |
| `start` | start position of subregion |
| `end` | end position of subregion |
| `chop.reads` | if TRUE reads will be choped at the boundaries of the region |

### Value

an epiG model

### Author(s)

Martin Vincent

### Examples

```
#TODO examples subregion
```

## `vector_search` *Pattern Search*

### Description

Search for pattern in integer vector

### Usage

```
vector_search(pattern, x)
```

### Arguments

| | |
|---|---|
| `pattern` | integer vector |
| `x` | integer vector to search in |

### Value

position of pattern in x

### Author(s)

Martin Vincent