AffyTiling

October 5, 2010

`AnalyzeTilingCelFiles`

Background correction and RNA normalization for CEL files from an Affymetrix tiling array.

Description

AnalyzeTilingCelFiles extracts intensity data from a group of CEL files and returns annotated intensities using information from a BPMAP file. Options can be set to limit the analysis to certain genomic features or regions of interest,thus requiring less memory and computing time.

By default, preforms background correction, quantile normalization, and log2 transform. This can be disabled by setting ReturnRawIntensities to TRUE, if raw intensity values are desired.

This function returns a matrix, where rows represent probes and columns represent the following values: –Unique probe ID –Probe start position (in genomic coordinates) –Chromosome –Sequence –Intensity for sample 1 –Intensity for sample 2 ... –Intensity for sample N

Usage

```
AnalyzeTilingCelFiles(CEL_filenames, BPMAP_filename, outfilename=NULL, iID=NULL,
```

Arguments

`CEL_filenames`	A character vector of the path to all CEL file(s) in the analysis.
`BPMAP_filename`	The path to the BPMAP file which describes the arrays specified in the cel files.
`outfilename`	If specified, the function writes a tab-separated table of normalized intensities.
`iID`	Vector of IDs for each interval specified.
`iCHR`	Vector of chromosomes for each interval.
`iSTART`	Integer vector of the interval start.
`iEND`	Integer vector of the interval end.
`makeUniqueID`	If TRUE (default), returns a column of unique identifiers for each probe, of the form: chr-start.
`readOnlyNCBI`	If TRUE (default), returns ONLY probes that target NCBI sequences, TIGR and Affymetrix controls are ignored.

`readProbeSeq` If TRUE, returns the first 25 bp of the probe sequence.

`IgnoreBpmapCelPlatformMismatch`
If TRUE, ignores a mismatch between BPMAP and CEL platforms. (EXPERT ONLY!)

`ReturnRawIntensities`
If TRUE, returns raw intensity values associated with the specified regions. Otherwise (default) preforms RMA-like processing of data using the affy package. Processing includes background correction, quantile normalization, and a log-2 transform.

Author(s)

Charles Danko

Examples

```
## Note that executing the following example requires .bpmap and .cel files in the workin
## If these files do not, the program will not execute.

## Get the file names in the current working directory.
CEL_NAMES <- dir(pattern=".CEL|.cel");
BPMAP     <- dir(pattern=".bpmap");

## If files are found in the current working directory ... start the analysis!!
if( (NROW(CEL_NAMES) > 0) & (NROW(BPMAP) > 0) ) {
  AnalyzeTilingCelFiles(CEL_NAMES, BPMAP, outfilename="NormalizedData.tsv");
}
```

AssociateWithGenes *Associates a vector of probe positions with the nearest input transcription start site.*

Description

Associates a vector of probes, with known genomic positions, to the nearest transcription start site (TSS).

If argument D is specified, returns ALL genes with a TSS < D bp from pID. Otherwise, the nearest gene is returned for each probe.

Returns vector composed of the following columns: ProbeID – Unique probe ID. GeneID – Gene/ transcript ID. pgDistance – Distance between ProbeID and GeneID TSS; Positive values indicate upstream of the TSS. Negative values indicate downstream.

Usage

```
AssociateWithGenes(kgID, kgCHR, kgSTR, kgSTART, kgEND, pID, pCHR, pMID, D=NULL)
```

Arguments

kgID	Vector of gene IDs, one for each transcription start site.
kgCHR	Vector of gene chromosomes, in the format "chr1", "chr2", ..., "chrX", "chrY".
kgSTR	Vector of strand: "+" for a gene on the "+1" strand, "-" for the "-1" strand.
kgSTART	Vector of start positions for each the transcript relative to the chromosome, NOT the start of transcription.
kgEND	Vector of end positions for each the transcript relative to the chromosome.
pID	Vector of UniqueIDs for each probe being evaluated.
pCHR	Vector of chromosomes, in the format "chr1", "chr2", ..., "chrX", "chrY".
pMID	Vector of start positions, one for each of the probes being annotated.
D	Threshold distance: if specified, returns all genes with a TSS < D bp from pID. Otherwise the nearest gene is returned for each probe.

Author(s)

Charles Danko

Examples

```
data(KnownGenes)

## Calculate the gene nearest to each probe in Einter data frame.
NearestGenes <- AssociateWithGenes(KG[,1], KG[,2], KG[,3], KG[,4], KG[,5],
    Einter[,1], Einter[,3], (as.integer(Einter[,2])+13))

## Returns all genes within 1 Kb of each probe in Einter.
## Probes that do not have a gene within 1 Kb are not returned.
NearestGenes <- AssociateWithGenes(KG[,1], KG[,2], KG[,3], KG[,4], KG[,5],
    Einter[,1], Einter[,3], (as.integer(Einter[,2])+13), D=1000)
```

AssociateWithInterval

Returns index of association between a probe and a set of genomic intervals.

Description

Takes four vectors representing a unique ID, chromosome, start, and end of non-overlapping regions of interest (e.g. known genes, CpG islands, etc.). Two additional vectors representing the chromosome and start position of probes in a tiling array.

One additional argument (optional) specifies the length of the probes. This can be either a vector with one length for each probe, or a scalar if all probes are the same length. If this argument is not specified, probes are assumed to be 25bp in length.

Returns a vector representing the unique ID of the interval in which each probe can be found. Values of NA are returned for probes that are located in any of the given genomic intervals.

Usage

```
AssociateWithInterval(fID, fCHR, fSTART, fEND, pCHR, pSTART, pLENGTH)
```

Arguments

fID	Vector of unique IDs, representing each interval searched.
fCHR	Vector of chromosomes, one for each interval searched.
fSTART	Vector of start positions, one for each interval searched.
fEND	Vector of end positions, one for the each interval searched.
pCHR	Vector of chromosomes, one for each probes being annotated.
pSTART	Vector of start positions, one for each of the probes being annotated.
pLENGTH	Vector or scalar representing probe length. Default is 25bp for each probe.

Author(s)

Charles Danko

Examples

```
data(KnownGenes)

## Returns probes that fall inside known genes.
Interval <- AssociateWithInterval( KG[,1], KG[,2], KG[,4], KG[,5], Einter[,3], as.integer
```

Einter *Sample intensities for human promoter tiling array.*

Description

Einter is a data frame consisting of 3 position information columns (Unique ID, Start position of probe, Chromosome) and normalized intensity data from 4 samples. Data is given for the first 3Mb of human chromosomes 1-3.

From the Affymetrix human promoter tiling array v02-3.

Full description is given below...

`UniqueID` A unique ID for each probe.

`START` The start position of each probe.

`CHR` The chromosome of each probe.

`X19.103.10009.CEL` Normalized, Log-2 transformed intensities for sample X19.103.10009.

`X19.104.10013.CEL` Normalized, Log-2 transformed intensities for sample X19.104.10013.

`X19.107.10027.CEL` Normalized, Log-2 transformed intensities for sample X19.107.10027.

`X19.104.10013.CEL` Normalized, Log-2 transformed intensities for sample X19.104.10013.

Usage

```
data(KnownGenes)
```

Format

data frame

Source

Unpublished data, courtesy of Yanli Zhang and Frank Middleton.

KG *Position of known genes in the first 2 Mb of human chromosome 1-3.*

Description

A data frame with 146 observations on the following 5 variables.

`name` Gene IDs.

`chrom` Chromosomes.

`strand` Strand.

`txStart` Transcription start.

`txEnd` Transcription end

Usage

```
data(KnownGenes)
```

Format

data frame

Source

The UCSC genome browser (http://genome.ucsc.edu).

`TilingCelFiles2Probesets`
Background correction and RNA normalization for CEL files from an Affymetrix tiling array.

Description

TilingCelFiles2Probesets extracts intensity data from a group of CEL files and returns annotated intensities using information from a BPMAP file. Options can be set to limit the analysis to certain genomic features or regions of interest,thus requiring less memory and computing time.

Returns a matrix, where rows represent probe sets and columns represent the following: –Unique probeset ID ("chromosome-first probe START position-last probe END position") –Probe start position (in genomic coordinates) –Chromosome –Average normalized intensity for sample 1 –Average normalized intensity for sample 2 ... –Average normalized intensity for sample N

Note that unlike AnalyzeTilingCelFiles, this function reports only 1 average value for all probes in each interval.

Usage

```
TilingCelFiles2Probesets(CEL_filenames, BPMAP_filename, outfilename=NULL, iID=NU
```

Arguments

CEL_filenames	A character vector of the path to all CEL file(s) in the analysis.
BPMAP_filename	The path to the BPMAP file which describes the arrays specified in the cel files.
outfilename	If specified, the function writes a tab-separated table of normalized intensities.
iID	Vector of IDs for each interval specified. If NULL (default) creates a unique ID for each interval of the form: "CHR-START-END".
iCHR	Vector of chromosomes for each interval.
iSTART	Integer vector of the interval start.
iEND	Integer vector of the interval end.
IgnoreBpmapCelPlatformMismatch	If TRUE, ignores a mismatch between BPMAP and CEL platforms. (EXPERT ONLY!)

Author(s)

Charles Danko

Examples

```
## Note that executing the following example requires .bpmap and .cel files in the workin
## If these files do not, the program will not execute.

## Creates a sample interval of the first 1MB of chromosome 1-3.
## This function will return a single value for each interval.
iCHR <- c("chr1", "chr2", "chr3")
iSTART <- rep(1, 3)
iEND <- iSTART + 1e+06

## Get the file names in the current working directory.
CEL_NAMES <- dir(pattern=".CEL|.cel");
BPMAP     <- dir(pattern=".bpmap");

## If files are found in the current working directory ... start the analysis!!
if( (NROW(CEL_NAMES) > 0) & (NROW(BPMAP) > 0) ) {
  TilingCelFiles2Probesets(CEL_NAMES, BPMAP, outfilename="NormalizedData.tsv", iID=NULL,
}
```

---

parseBPMAP *Returns information from a BPMAP file for select probes in an Affymetrix tiling array.*

Description

parseBPMAP takes as input the path to an Affymetrix BPMAP file, and data on any genomic intervals of particular interest to the analysis.

Returns a matrix, where rows represent each probe on the tiling array and columns representing the following information: "UniqueID" – A unique ID for each probeset of the form: "Chromosome-Start" (if makeUniqueID == TRUE) "CHR" – Chromosome on which the probe is located "Start"

– Genomic position in the BPMAP file "PMX" – X index of the spot on the tiling array "PMY" – Y index of the spot on the tiling array "SEQ" – Sequence of the probe (if readProbeSeq == TRUE) "iID" – The ID of the interval in which this probe is located (if an interval was passed)

Note that passing a region of interest returns only probes in that genomic region. If no region is specified, information is returned for all probes on the tiling array.

Usage

```
parseBPMAP(filename, iID=NULL, iCHR=NULL, iSTART=NULL, iEND=NULL, recordInterval
```

Arguments

`filename`	The path to the BPMAP file which describes the arrays specified in the cel files.
`iID`	Vector of IDs for each interval specified.
`iCHR`	Vector of chromosomes for each interval.
`iSTART`	Integer vector of the interval start.
`iEND`	Integer vector of the interval end.
`recordIntervalIDs`	If TRUE, returns a column of the interval ID corresponding to each probe. Requires use of interval data.
`makeUniqueID`	If TRUE (default), returns a column of unique identifiers for each probe, of the form: "chr"-"start"
`readOnlyNCBI`	If TRUE (default), returns ONLY probes that target NCBI sequences, TIGR and Affymetrix controls are ignored.
`seqIndices`	If specified, reads only given portions of the BPMAP file (Expert ONLY).
`readProbeSeq`	If TRUE, returns the first 25 bp of the probe sequence.
`verbose`	if >= 1, returns varying amounts of output in the R window.

Author(s)

Charles Danko

Examples

```
## Note that executing the following example requires a .bpmap file in the working direct
## If one does not exist, the program will not execute.

## Identify a .bpmap file in the current working directory.
BPMAP <- dir(pattern=".bpmap");

## If one or more .bpmap file are present in the current working directory
## returns a list representation of information in the first .bpmap file.
if( NROW(BPMAP) > 0 ) {
  parseBPMAP(BPMAP[0]);
}
```

Index