Package 'NanoR'

March 25, 2019

Type Package

Title Analyze Nanopore MinION and GridION X5 1D Sequencing Data

Version 2.0.0

Author Davide Bolognini, BS, PhD Fellow [aut, cre]

Maintainer Davide Bolognini <davidebolognini7@gmail.com>

Depends R (>= 3.1.3)

Imports parallel,
ggplot2 (>= 2.2.1),
reshape2,
scales,
RColorBrewer,
grid,
gridExtra,
rhdf5(>= 2.14),
ShortRead (>= 1.24.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

NanoCompare *Compare ONT experiments*

Description

NanoCompare fast build violin plots, comparing MinION and GridION X5 experiments analyzed with the other functions from this package. Bins of 30 minutes are taken.

Usage

```
NanoCompare(DataIn, DataOut, Labels)
```

Arguments

DataIn	Character vector with paths to "DataForComparison" for each experiment
DataOut	Where results will be saved.
Labels	Character vector containing ordered labels to name experiments in "DataIn"

Value

Plot:
- Violins.pdf;

Examples

```
#do not run
DataIn<-c("Path/To/AnalyzedFolder1/DataForComparison","Path/To/AnalyzedFolder2/DataForComparison",...)
Labels<-c("Label1","Label2") #labels
NanoCompare(DataIn=DataIn,DataOut="Path/To/DataOut",Labels=Labels) #compare
```

NanoFastqG *Filter .fastq files*

Description

Retain only high-quality .fastq from .fastq files (optionally, converts .fastq to .fasta as well).

Usage

```
NanoFastqG(DataSummary, DataFastq, DataOut, Label, Cores = 1, FASTA = FALSE,
  Minquality = 7)
```

Arguments

DataSummary	Path to sequencing summary file/s folder
DataFastq	Path to passed .fastq folder
DataOut	Where the .fastq file will be saved
Label	Label to identify the experiment. A folder with this name will be created in DataOut directory.
Cores	Number of cores to use to accelerate sequencing summary files reading when dealing with multiple ones. Default to 1
FASTA	Logical. If TRUE translate .fastq to .fasta as well. Default to FALSE
Minquality	Minimum quality to retain the .fastq sequence. Default to 7
FASTQTOT	Logical. If TRUE, combine all the .fastq together and store in the DataOut folder. Default to FALSE. #removed because there are faster way to concatenate. In bash, use "cat".

Value

filtered .fastq file and, optionally, convert to .fastq file

Examples

```
#do not run
new behaviour
DataSummary<-"/path/to/sequencing_summary"
DataFastq<-"/path/to/fastq_pass"
DataOut <- "/path/to/DataOut"
Label<-'Exp'
Filter on higher treshold
NanoFastqG(DataSummary, DataFastq, DataOut, Label, Minquality=10)
At the moment, NanoFastqM do not offer gz compression, as is it very slow to do in the R environment.
```

NanoFastqM *Extracts .fastq informations*

Description

NanoPrepareM prepares MinION and GridION X5 basecalled data for other functions from this package. Name of the function is inherited from NanoR previous version as MinION and GridION X5 had different default-output formats and only MinION outputted basecalled .fast5 files. From MinION release 18.12 and GridION 18.12.1 outputs are the same. This is useful expecially to filter at different treshold than the default one.

Usage

```
NanoFastqM(DataPass, DataOut, Label, Cores = 1, FASTA = FALSE,
  Minquality = 7, MultiRead = FALSE)
```

Arguments

DataPass	Path to passed .fast5 files folder
DataOut	Where the .fastq file will be saved
Label	Label to identify the MinION experiment. A folder with this name will be created in DataOut directory.
Cores	Number of cores to be used: 1 by default
FASTA	Logical. If TRUE translate .fastq to .fasta as well. Default to FALSE
Minquality	Minimum quality to retain the .fastq sequence. Default to 7
MultiRead	Logical. If TRUE, enable multiread .fast5 files support. Default to FALSE.

Value

.fastq file and, optionally, .fasta file for MinION passed .fast5 files

Examples

```
#do not run
DataPass<-"/path/to/fast5_pass"
DataOut <- "/path/to/DataOut"
Label<-'Exp'
#single-read .fast5 files
#Extract
NanoFastqM(DataPass, DataOut, Label, Cores=6) #Minquality=7
#Extract, convert to .fastq and filter.
NanoFastqM(DataPass, DataOut, Label, Cores=6, FASTA=TRUE, Minquality=10)
Extract from multi-read .fast5 files
NanoFastqM(DataPass, DataOut, Cores=6, MultiRead=TRUE)
At the moment, NanoFastqM do not offer gz compression, as is it very slow to do in the R environment.
```

NanoPrepareG *Prepares sequencing summary and .fastq files*

Description

NanoPrepareG prepares MinION and GridION X5 sequencing summary and .fastq files for other functions from this package. Name of the function is inherited from NanoR previous version as MinION and GridION X5 had different default-output formats and only GridION X5 outputted sequencing summary and .fastq files. From MinION release 18.12 and GridION 18.12.1 outputs are the same.

Usage

```
NanoPrepareG(DataSummary, DataFastq, Cores = 1, Label)
```

Arguments

DataSummary	Path to sequencing summary file/s folder
DataFastq	Path to passed .fastq folder.
Cores	Number of cores to accelerate sequencing summary files reading when dealing with multiple ones. Default to 1
Label	Label to identify the experiment. A folder with this name will be created in DataOut directory.

Details

NanoPreareG can find desired input files recursively, so be careful to specify path to folder contaning a unique data type. Old releases of GridION X5 stored all the .fastq files (passed and failed) in the same directory. In this case, this directory can be given as "DataFastq" input.

Value

Object of class list

Examples

```
#do not run
DataSummary<-'path/to/sequencing_summary'
DataFastq<-'path/to/fastq_pass'
Label<-'Exp'
#new behaviour
List<-NanoPrepareG(DataSummary, DataFastq, Label=Label)
#old behaviour (same folder for DataSummary and DataFastq)
Data<-'path/to/sequencing_summary'<-'path/to/fastq_pass'
List<-NanoPrepareG(Data, Data, Cores=5, Label=Label)
```

NanoPrepareM *Prepares basecalled data*

Description

NanoPrepareM prepares MinION and GridION X5 basecalled data for other functions from this package. Name of the function is inherited from NanoR previous version as MinION and GridION X5 had different default-output formats and only MinION outputted basecalled .fast5 files. From MinION release 18.12 and GridION 18.12.1 outputs are the same.

Usage

```
NanoPrepareM(DataPass, DataFail = NA, DataSkip = NA, Label,
  MultiRead = FALSE)
```

Arguments

DataPass	Path to passed .fast5 files folder
DataFail	Path to failed .fast5 files folder. Default to NA
DataSkip	Path to skipped .fast5 files folder. Default to NA
Label	Label to identify the experiment. A folder with this name will be created in DataOut directory
MultiRead	Logical. If TRUE, enable multi-read .fast5 files support. Default to FALSE

Details

NanoPreareM can find .fast5 files recursively, so be careful to specify path to folder contaning a unique data type (e.g. only passed .fast5 files for the experiment). DataFail and DataSkip can be omitted.

Value

Object of class list

Examples

```
#do not run
DataPass<-"/path/to/fast5_pass"
DataFail<-"/path/to/fast5_fail" #can be omitted
# PathSkip. Useful for old MinION data versions
Label<-"Exp"
#single-read .fast5 files
List<-NanoPrepareM(DataPass,DataFail,Label=Label)
#multi-read .fast5 files
List<-NanoPrepareM(DataPass,DataFail, Label=Label,MultiRead=TRUE)
```

NanoStatsG *Plots statistics*

Description

NanoStatsG plots statistics parsing the metadata table returned by NanoTableG

Usage

```
NanoStatsG(NanoGList, NanoGTable, DataOut, KeepGGObj = FALSE)
```

Arguments

NanoGList	Object of class list returned by NanoPrepareG
NanoGTable	Metadata table returned by NanoTableG
DataOut	Where results will be saved. Use the same directory specified as "DataOut" for NanoTableG
KeepGGObj	Store data.frames for ggplot plots in folder. Useful for personalize plot colors. Default to FALSE.

Value

Plots:
- Yield.pdf (accumulation of reads and bps);
- RBLQ.pdf (# reads, # bps, length and quality overview every 30 minutes of experiment);
- LvQ.pdf (length and quality compared jointly);
- PFGC.pdf (passed and failed reads, GC content if previously computed);
- Activity.pdf (channels activity (# bps). Inactive channels are grey-colored)
Tables:
- metadata.fltrd.txt (metadata table for high-quality passed sequence) - ShortSummary.txt (table with major statistics for the experiment)

Examples

```
DataOut <- "/path/to/DataOut"
# Need a list previously generated with NanoPreparGM()
# Need a table previously generated with NanoTableG()
# If List from NanoPrepareG() and Table from NanoTableG() exist:
# Do not save ggplot2 tables:
NanoStatsG(List,Table, DataOut=DataOut)
# Save ggplot2 tables:
NanoStatsG(List,Table, DataOut=DataOut,KeepGGObj=TRUE)
```

NanoStatsM *Plots statistics*

Description

NanoStatsM plots statistics parsing the metadata table returned by NanoTableM

Usage

```
NanoStatsM(NanoMList, NanoMTable, DataOut, KeepGGObj = FALSE)
```

Arguments

NanoMList	Object of class list returned by NanoPrepareM
NanoMTable	Metadata table returned by NanoTableM
DataOut	Where results will be saved. Use the same directory specified as "DataOut" for NanoTableM
KeepGGObj	Store data.frames for ggplot plots in folder. Useful for personalize plot colors. Default to FALSE.

Value

Plots:
- Yield.pdf (accumulation of reads and bps);
- RBLQ.pdf (# reads, # bps, length and quality overview every 30 minutes of experiment);
- LvQ.pdf (length and quality compared jointly);
- PFGC.pdf (passed and failed reads, GC content if previously computed);
- Activity.pdf (channels and muxes activity (# bps). Inactive channels and muxes are grey-colored)
Tables:
- metadata.fltrd.txt (metadata table for high-quality passed sequence) - ShortSummary.txt (table with major statistics for the experiment)

Examples

```
#do not run
DataOut <- "/path/to/DataOut"
# Need a list previously generated with NanoPrepareM()
# Need a table previously generated with NanoTableM()
# If List from NanoPrepareM() and Table from NanoTableM() exist:
# Do not save ggplot2 tables:
NanoStatsM(List,Table, DataOut=DataOut)
# Save ggplot2 tables:
NanoStatsM(List,Table, DataOut=DataOut,KeepGGObj=TRUE)
```

`NanoTableG` *Generates metadata table*

Description

NanoTableG filters the sequencing summary table retaining only the most-useful statistics and optionall extract GC content from .fastq files

Usage

```
NanoTableG(NanoGList, DataOut, GCC = FALSE)
```

Arguments

`NanoGList`	Object of class list returned by NanoPrepareG
`DataOut`	Where the metadata table will be saved. Do not use a directory that already contains other results.
`GCC`	Logical. If TRUE, NanoTableG computes GC content for each sequence in .fastq files. Default to FALSE. Calculating GCC can be slow in R.

Value

Metadata table with 7 columns

Examples

```
#do not run
DataOut <- "/path/to/DataOut"
# Need a list previously generated with NanoPrepareG()
# If List from NanoPrepareM() exists:
# Skip GC content calculation
Table<-NanoTableG(List, DataOut)
# Calculate GC content
Table<-NanoTableG(List, DataOut, GCC=TRUE)
```

NanoTableM *Generates metadata table*

Description

NanoTableM generates a table that contains key informations for each passed .fast5 read. As NanoTableM can take some time for large sets, this function can be accelerated using multiple cores

Usage

```
NanoTableM(NanoMList, DataOut, Cores = 1, GCC = FALSE)
```

Arguments

NanoMList	Object of class list returned by NanoPrepareM
DataOut	Where the metadata table will be saved. Do not use a directory that already contains other results.
Cores	Number of cores to accelerate metadata extraction: 1 by default
GCC	Logical. If TRUE, NanoTableM computes GC content for each read. Default to FALSE

Value

Metadata table with 7 columns

Examples

```
#do not run
DataOut <- "/path/to/DataOut"
# Need a list previously generated with NanoPrepareM()
# If List from NanoPrepareM() exists:
# Skip GC content calculation
Table<-NanoTableM(List,DataOut,Cores=6)
# Calculate GC content
Table<-NanoTableM(List,DataOut,Cores=6, GCC=TRUE)
```

Index