Package

February 14, 2019

Type Package

Title Analyze and comapre ONT MinION and GridION X5 1D Sequencing Data

Version 1.0

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),
seqinr,
reshape2,
scales,
RColorBrewer,
grid,
gridExtra,
rhdf5(>= 2.14),
ShortRead (>= 1.24.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

FastqFilterG *Filter your GridION X5 .fastq files*

Description

Use this function if you have .fastq and sequencing summary files and you want to filter your .fastq files in order to obtain only the high-quality ones. FastqFilterG can optionally return a .fasta file for the high-quality sequences. FastqFilterG can optionally return a total .fastq file (equivalent to a 'cat' shell command) and translate into .fasta

Usage

```
FastqFilterG(Data, DataOut, FASTQTOT = FALSE, FASTA = FALSE, Cores = 1,
  Label, Minquality = 7)
```

Arguments

Data	Path to .fastq and sequencing summary files returned from GridION X5 (can find .fastq and sequencing summary files recursively)
DataOut	Where the .fastq (and, optionally, .fasta) file will be saved
FASTQTOT	Logical. If TRUE, combine all the .fastq together and store in the DataOut folder. Default to FALSE
FASTA	Logical. If FALSE, return only .fastq file else, if TRUE, return both .fastq and .fasta files. Default to FALSE
Cores	Number of cores to use to accelerate sequencing summary files reading. Default to 1
Label	Label to use, together with the Flow Cell identifier, to identify the experiment
Minquality	Minimum quality to retain the .fastq sequence. Default to 7

Value

High-quality .fastq file and, optionally, high quality .fasta file, total .fastq file and total .fasta file. If one or more .fastq file is "ill-formatted", FastqFilterG stops and prints the number of the "guilty" .fastq file.

Examples

```
#do not run
DataPath<-"/data/basecalled/ExperimentName/FlowCellId"
FastqFilterG(Data=DataPath, DataOut="Path/To/DataOut",FASTQTOT=FALSE,FASTA=FALSE)
```

NanoCompare *Compare ONT experiments*

Description

NanoCompare plots comparison statistics for MinION and GridION X5 experiments analyzed with the other functions from this package

Usage

```
NanoCompare(DataIn, DataOut, Labels, GCC = TRUE)
```

Arguments

DataIn	Character vector containing paths to folders contaning analyzed MinION and/or GridION X5 experiments
DataOut	Where to save NanoCompare results
Labels	Character vector containing ordered labels used to identify experiments in DataIn

Value

Plots:
- Violins.pdf;
- Histograms.pdf;

Examples

```
#do not run
DataIn<-c("Path/To/AnalyzedFolder1","Path/To/AnalyzedFolder2",...)
Labels<-c("Label1","Label2","Label3") #labels used
NanoCompare(DataIn=DataIn,DataOut="Path/To/DataOut",Labels=Labels,GCC=TRUE) #compare
```

NanoFastqG *Extracts .fastq informations from your GridION X5 basecalled passed .fast5 files*

Description

NanoFastqG returns a .fastq file (and, optionally, a .fasta file) for your high-quality reads

Usage

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

Arguments

DataPass	Path to passed .fast5 files folder
DataOut	Where .fastq (and, optionally, .fasta) file will be saved
Label	Label used, together with the Flow Cell identifier extracted from the inputted data, to identify .fastq (and, optionally, .fasta) file
Cores	Number of cores to be used: 1 by default
FASTA	Logical. If FALSE, return only .fastq file else, if TRUE, return both .fastq and .fasta files. Default to FALSE
Minquality	Minimum quality to retain the .fastq sequence. Default to 7

Value

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

Examples

```
#do not run
NanoFastqG(DataPass="Path/To/DataPass", DataOut="/Path/To/DataOutExp", Cores=6, FASTA=FALSE)
NanoFastqG(DataPass="Path/To/DataPass", DataOut="/Path/To/DataOutExp", Cores=6, FASTA=TRUE)
```

NanoFastqM *Extracts .fastq informations from your MinION passed .fast5 files*

Description

NanoFastqM returns a .fastq file (and, optionally, a .fasta file) for your high-quality reads

Usage

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

Arguments

DataPass	Path to passed .fast5 files folder
DataOut	Where .fastq (and, optionally, .fasta) file will be saved
Label	Label used to identify .fastq (and, optionally, .fasta) file
Cores	Number of cores to be used: 1 by default
FASTA	Logical. If FALSE, return only .fastq file else, if TRUE, return both .fastq and .fasta files. Default to FALSE
Minquality	Minimum quality to retain the .fastq sequence. Default to 7

Value

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

Examples

```
#do not run
NanoFastqM(DataPass="Path/To/DataPass", DataOut="/Path/To/DataOutExp", Cores=6, FASTA=FALSE)
NanoFastqM(DataPass="Path/To/DataPass", DataOut="/Path/To/DataOutExp", Cores=6, FASTA=TRUE)
```

`NanoPrepareG` *Prepares GridION X5 data for your analyses with NanoR*

Description

NanoPrepareG generates an object of class list that contains informations required by other functions from NanoR when analyzing GridION X5 data.

Usage

```
NanoPrepareG(BasecalledFast5 = FALSE, Data, DataFail = NA, DataSkip = NA,
  Cores = 1, Label)
```

Arguments

`BasecalledFast5`	Logical. TRUE if dealing with basecalled .fast5 files. Defaulto to FALSE
`Data`	Path to GridION X5 folder containing .fastq and sequencing summary files (if BasecalledFast5 = FALSE) or to basecalled .fast5 files (if BasecalledFast5 = TRUE)
`DataFail`	Path to failed .fast5 files folder
`DataSkip`	Path to skipped .fast5 files folder
`Cores`	Number of cores to be used to accelerate sequencing summary files reading (useful only if BasecalledFast5 = FALSE)
`Label`	Label used, together with the Flow Cell identifier extracted from the inputted data, to identify your experiment: do not use underscore characters ("_").

Details

NanoPrepareg can find desired inputs recursively. DataSkip and DataFail can be omitted.

Value

Object of class list containing informations required by NanoTableG and NanoStatsG functions.

Examples

```
#do not run
#when working with sequencing summary files and .fastq files
Data<-"/data/basecalled/ExperimentName/FlowCellId"
NanoGList<-NanoPrepareG(BasecalledFast5=FALSE, Data=Data, Label="Ex", Cores=3)
#when working with basecalled .fast5 files
Pass<-"Path/to/workspace/pass"
Fail<-"Path/to/workspace/fail"
NanoGList<-NanoPrepareG(BasecalledFast5=TRUE, Data=Pass, DataFail=Fail, Label="Ex")
```

NanoPrepareM *Prepares MinION data for your analyses with NanoR*

Description

NanoPrepareM generates an object of class list that contains informations required by other functions from NanoR when analyzing MinION data

Usage

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

Arguments

DataPass	Path to MinION passed .fast5 files folder
DataFail	Path to MinION failes .fast5 files folder
DataSkip	Path to MinION skipped .fast5 files folder
Label	Label to identify your MinION experiment

Details

NanoPreareM can find .fast5 files recursivel. DataFail and DataSkip can be omitted (MinKNOW generates passed, failes and skipped .fast5 files folders but failed and skipped .fast5 files are taken into account only for calculating their number and percentage)

Value

Object of class list

Examples

```
#do not run
PathPass<-"/Path/To/PassFast5"
Lab<-"Exp"
NanoMList<-NanoPrepareM(DataPass=PathPass, Label=Lab)
```

NanoStatsG *Plots statistics for your GridION X5 .fast5 files*

Description

NanoStatsG plots statistics for passed .fast5 files and returns 4 tables used by NanoCompare

Usage

```
NanoStatsG(NanoPrepareGList, NanoGTable, DataOut)
```

Arguments

NanoPrepareGList	Object of class list returned by NanoPrepareG
NanoGTable	Table returned by NanoTableG
DataOut	Where NanoStatsG results will be saved. Use the same directory specified for NanoTableG function and be sure that it doesn't already contain NanoStatsM (or NanoStatsG) results

Value

Plots:
- Cumulative_Reads_&_Cumulative_Basepairs.pdf;
- Reads_Basepairs_Length_Quality.pdf;
- Length_versus_Quality.pdf;
- Pass_Fail_Skip_and_GC_Content.pdf or Pass_Fail_Skip_NO_GC_Content.pdf;
- Channels_Activity.pdf or Channels_and_Muxes_Activity.pdf. Not-working channels and muxes are grey-colored.

Examples

```
#do not run
#knows how to deal with different inputs type autonomously
NanoStatsG(NanoPrepareGList=NanoGList, NanoGTable=NanoGTable, DataOut="/Path/To/DataOutEx")
```

NanoStatsM *Plots statistics for your MinION .fast5 files*

Description

NanoStatsM plots statistics for passed .fast5 files and returns 4 tables used by NanoCompare

Usage

```
NanoStatsM(NanoPrepareMList, NanoMTable, DataOut)
```

Arguments

NanoPrepareMList	Object of class list returned by NanoPrepareM
NanoMTable	Table returned by NanoTableM
DataOut	Where NanoStatsM results will be saved. Use the same directory specified for NanoTableM function and be sure that it doesn't already contain NanoStatsM (or NanoStatsG) results

Value

Plots:
- Cumulative_Reads_&_Cumulative_Basepairs.pdf;
- Reads_Basepairs_Length_Quality.pdf;
- Length_versus_Quality.pdf;
- Pass_Fail_Skip_and_GC_Content.pdf or Pass_Fail_Skip_NO_GC_Content.pdf;
- Channels_and_Muxes_Activity.pdf. Inactive channels and muxes are grey-colored

Examples

```
#do not run
NanoStatsM(NanoPrepareMList=NanoMList, NanoMTable=NanoMTable, DataOut="/Path/To/DataOutExp")
```

NanoTableG *Generates an information table for your GridION X5 .fast5 files*

Description

NanoTableG generates a table that contains useful informations for each read identified by NanoPrepareG function. When analyzing GridION X5 basecalled .fast5 files, metadata extraction can be accelerated using multiple cores. Sometimes .fastq files returned by GridION X5 can be ill-formatted: in this case, NanoTableG will stop and you can run this function again after set GCC to FALSE. This problem can be avoided if basecalled .fast5 files are used.

Usage

```
NanoTableG(NanoPrepareGList, DataOut, Cores = 1, GCC = TRUE)
```

Arguments

NanoPrepareGList	Object of class list returned by NanoPrepareG function
DataOut	Where the table will be saved. Do not use a directory that already contains a NanoTableM (or NanoTableG) result
Cores	Number of cores to be used: 1 by default. Does not affect time when dealing with .fastq and sequencing summary files.
GCC	Logical. If TRUE, NanoTableM computes GC content for each read. Default to TRUE

Value

Table with 7 columns

Examples

```
#do not run
DataOut <- "/Path/To/DataOutEx"
#when working with sequencing summary files and .fastq files
NanoGTable<-NanoTableG(NanoPrepareGList=NanoGList, DataOut=DataOut) #set GCC to "FALSE" on error
#when working with basecalled .fast5 files
NanoGTable<-NanoTableG(NanoPrepareGList=NanoGList, DataOut=DataOut, GCC=TRUE, Cores=6)
```

NanoTableM *Generates a metadata table for your MinION .fast5 files*

Description

NanoTableM generates a table that contains useful informations for each read of the "pass" .fast5 files folder given to NanoPrepareM function. As NanoTableM can take some time (it depends on the number of reads it has to deal with), this function can be accelerated using multiple cores

Usage

```
NanoTableM(NanoPrepareMList, DataOut, Cores = 1, GCC = TRUE)
```

Arguments

NanoPrepareMList	Object of class list returned by NanoPrepareM
DataOut	Where the table will be saved. Do not use a directory that already contains a NanoTableM (or NanoTableG) result
Cores	Number of cores to be used: 1 by default
GCC	Logical. If TRUE, NanoTableM computes GC content for each read. Default to TRUE

Value

Table with 7 columns

Examples

```
#do not run
NanoMTable<-NanoTableM(NanoPrepareMList=NanoMList, DataOut="/Path/To/DataOutExp",Cores=6,GCC=TRUE)
```

Index