API Documentation

API Documentation

October 5, 2016

Contents

1 Script script-iReportMaker

Make a PDF report from the log file generated by the VarCall workflow.

1.1 Functions

---

get_parser()

Parse arguments

Return Value
> arguments list
>
> *(type=parser object)*

---

texHeader(*texFile*)

Write the header of the LaTeX output file

Parameters
> `texFile`: path of the output LaTeX file
>
> *(type=string)*

---

writeVarCallCommand(*lines*, *texFile*)

Write the VarCall command in the LaTeX output file

Parameters
> `lines`: lines of the log file
>
> *(type=list)*
>
> `texFile`: path of the output LaTeX file
>
> *(type=string)*

---

extractReadStatistics(*lines*)

Extract reads statistics from the log file lines.

Parameters
> `lines`: lines of the log file
>
> *(type=list)*

Return Value
> dictionnary with sample name in key and for value an other dictionnary with the number of reads before and after trimming, the number of reads mapped and properly paired, and the deep and breadth coverage.
>
> *(type=dictionnary)*

---

workdir()

Find working directory and transformed it for write in latex file

Return Value
    working directory transformed

    (type=string)

---

version()

Find all version of all tools used by VarCall

Return Value
    dictionnary with program name as key and version as value.

    (type=dictionnary)

---

main()

1.2 Variables

Name	Description
__doc__	**Value:** ...
__package__	**Value:** None

2 Script script-iVARCall

Do the SNP calling from several FASTQ files

Requires:
- BWA
- samtools
- picard-tools
- Trimmomatic
- GATK
- SnpSift
- BAMmaker, iVCFmaker_SNP, iVCFmerge_SNP, VCFilter, VCFtoMATRIX and VCFto-FASTA in $PATH

    Paired-end reads must be nammed as "prefix_R1.XXXX" and "prefix_R2.XXXX". Example : E1_R1.fq and E1_R2.fq

2.1 Functions

---

get_parser(*)

Parse arguments

Return Value
    arguments list

    (type=parser object)

---

checkPE(*readsList*)

checks all paired-end reads and reordering them

Parameters
    readsList: list of reads given by the argument parser

                (type=list)

Return Value
    read list

    (type=list)

---

checkFiles(*fileList*)

Check if all file of a list exist

Parameters
    fileList: list file pathway

                (type=list)

---

commandLaunched(*Arguments, logFile*)

Print VarCall command in log file

Parameters

    `Arguments:` list of arguments

                    (type=parser object)

    `logFile:`   log file

                    (type=string)

---

main()

2.2   Variables

Name	Description
__doc__	**Value:** ...
__package__	**Value:** `None`
accept2dyear	**Value:** `1`
altzone	**Value:** `-7200`
daylight	**Value:** `1`
timezone	**Value:** `-3600`
tzname	**Value:** `('CET', 'CEST')`

3   Script script-iVCFmaker_SNP

Requires: GATK

3.1   Functions

---
get_parser()

Parse arguments

Return Value
     arguments list

     (type=parser object)

---
exec_commands(*cmds, nbThreads*)


```
Exec commands in parallel in multiple process (as much as we have CPU)
@param cmds: list of commands
    @type cmds: list
    @param nbThreads: number of threads to use
    @type nbThreads: integer
```

---
main()

3.2   Variables

Name	Description
__doc__	**Value:** ...
__package__	**Value:** None

4 Script script-iVCFmerge_SNP

Requires:

- bgzip
- tabix
- merge-vcf

4.1 Functions

---

get_parser()

Parse arguments

Return Value
    arguments list

    (type=parser object)

---

exec_commands(*cmds*, *nbThreads*)

```
Exec commands in parallel in multiple process (as much as we have CPU)
@param cmds: list of commands
    @type cmds: list
    @param nbThreads: number of threads to use
    @type nbThreads: integer
```

---

main()

4.2 Variables

Name	Description
__doc__	**Value:** ...
__package__	**Value:** None

Index