## FastqArazketa

Generated by Doxygen 1.8.8

Wed Aug 9 2017 16:09:38

# **Contents**

1	Clas	s Index																1
	1.1	Class I	_ist					 	 		 		 				 	1
2	File	Index																3
	2.1	File Lis	st					 	 		 	-	 				 	3
3	Clas	s Docu	mentation	n														5
	3.1	_fq_rea	ad Struct F	Refer	ence			 	 		 		 				 	5
		3.1.1	Detailed	Desc	cription	١		 	 		 		 				 	5
		3.1.2	Member	Data	Docu	menta	ation	 	 		 		 				 	5
			3.1.2.1	L.				 	 		 		 				 	5
			3.1.2.2	sta	rt			 	 		 		 				 	5
	3.2	_iparar	m_Qreport	t Stru	ıct Ref	erenc	e .	 	 		 		 				 	5
		3.2.1	Detailed	Desc	cription	ı		 	 		 		 				 	6
		3.2.2	Member	Data	Docu	menta	ation	 	 		 		 				 	6
			3.2.2.1	filte	er			 	 		 		 				 	6
			3.2.2.2	inp	utfile .			 	 		 		 				 	6
			3.2.2.3	mir	nQ			 	 		 		 				 	6
			3.2.2.4	nQ				 	 		 		 				 	6
			3.2.2.5	ntile	es			 	 		 		 				 	6
			3.2.2.6	one	e_read	_len		 	 		 		 				 	6
			3.2.2.7	out	putfilel	bin .		 	 		 		 				 	6
			3.2.2.8	out	putfilel	html .		 	 		 		 				 	6
			3.2.2.9	out	putfilei	info .		 	 		 		 				 	7
			3.2.2.10	rea	d_len			 	 		 		 				 	7
	3.3	_iparar	m_Sreport	t Stru	ct Ref	erenc	e .	 	 		 		 				 	7
		3.3.1	Detailed	Desc	cription	ı		 	 		 		 				 	7
		3.3.2	Member	Data	Docu	menta	ation	 	 		 		 				 	7
			3.3.2.1	inp	utfolde	er		 	 		 		 				 	7
			3.3.2.2		putfile													
	3.4	statsin	fo Struct R		•													
	-		Detailed					-		-	-						-	8

iv CONTENTS

		3.4.2	Member	Data Documentation	8
			3.4.2.1	ACGT_pos	8
			3.4.2.2	ACGT_tile	8
			3.4.2.3	lane_tags	8
			3.4.2.4	lowQ_ACGT_tile	8
			3.4.2.5	minQ	8
			3.4.2.6	nQ	8
			3.4.2.7	nreads	9
			3.4.2.8	ntiles	9
			3.4.2.9	QPosTile_table	9
			3.4.2.10	qual_tags	9
			3.4.2.11	read_len	9
			3.4.2.12	reads_MlowQ	9
			3.4.2.13	reads_wN	9
			3.4.2.14	sz_ACGT_pos	9
			3.4.2.15	sz_ACGT_tile	9
			3.4.2.16	sz_lowQ_ACGT_tile	9
			3.4.2.17	sz_QPosTile_table	9
			3.4.2.18	sz_reads_MlowQ	9
			3.4.2.19	tile_pos	10
			3.4.2.20	tile_tags	10
4	File I	Docume	entation		11
	4.1			File Reference	11
		4.1.1		Description	
				efinition Documentation	12
			4.1.2.1	DEFAULT MINQ	12
			4.1.2.2	DEFAULT_NQ	12
			4.1.2.3	DEFAULT NTILES	12
			4.1.2.4	MAX FILENAME	12
			4.1.2.5	MAX_RCOMMAND	12
			4.1.2.6	N_ACGT	12
			4.1.2.7	ZEROQ	12
	4.2	include	/fopen_ge	n.h File Reference	12
		4.2.1	Detailed	Description	13
		4.2.2	Function	Documentation	14
			4.2.2.1	fopen_gen	14
	4.3	include	/fq_read.h	File Reference	14
		4.3.1	Detailed	Description	15
		4.3.2	Function	Documentation	15

CONTENTS

		4.3.2.1	get_fqread			 	 	 	15
		4.3.2.2	string_seq			 	 	 	15
4.4	include	/init_Qrepo	ort.h File Refere	nce		 	 	 	16
	4.4.1	Detailed I	Description			 	 	 	17
4.5	include	/Rcommar	nd_Qreport.h Fil	e Referen	ice	 	 	 	17
	4.5.1	Detailed I	Description			 	 	 	17
4.6	include	/Rcommar	nd_Sreport.h Fil	e Referen	ce	 	 	 	18
	4.6.1	Detailed I	Description			 	 	 	18
4.7	include	/stats_info	h File Referenc	e		 	 	 	18
	4.7.1	Detailed I	Description			 	 	 	20
	4.7.2	Function	Documentation			 	 	 	20
		4.7.2.1	init_info			 	 	 	20
		4.7.2.2	resize_info			 	 	 	20
		4.7.2.3	update_ACGT	_counts .		 	 	 	20
4.8	include	/str_manip	.h File Reference	е		 	 	 	21
	4.8.1	Detailed I	Description			 	 	 	21
4.9	src/fope	en_gen.c F	File Reference .			 	 	 	21
	4.9.1	Detailed I	Description			 	 	 	22
	4.9.2	Function	Documentation			 	 	 	23
		4.9.2.1	compress			 	 	 	23
		4.9.2.2	fcompress			 	 	 	23
		4.9.2.3	fopen_gen			 	 	 	23
		4.9.2.4	funcompress .			 	 	 	23
		4.9.2.5	uncompress .			 	 	 	23
4.10	src/fq_i	read.c File	Reference			 	 	 	23
	4.10.1	Detailed I	Description			 	 	 	24
	4.10.2	Function	Documentation			 	 	 	24
		4.10.2.1	get_fqread			 	 	 	24
		4.10.2.2	string_seq			 	 	 	25
	4.10.3	Variable I	Documentation			 	 	 	25
		4.10.3.1	par_QR			 	 	 	25
4.11	src/init_	_Qreport.c	File Reference			 	 	 	25
	4.11.1	Detailed I	Description			 	 	 	26
	4.11.2	Variable I	Documentation			 	 	 	26
		4.11.2.1	par_QR			 	 	 	26
4.12	src/init_	_Sreport.c	File Reference			 	 	 	26
	4.12.1	Detailed I	Description			 	 	 	27
	4.12.2	Variable I	Documentation			 	 	 	27
		4.12.2.1	par_SR			 	 	 	27
4.13	src/Qre	port.c File	Reference			 	 	 	27

vi CONTENTS

	4.13.1	Detailed Description	28
	4.13.2	Macro Definition Documentation	28
		4.13.2.1 B_LEN	28
	4.13.3	Variable Documentation	28
		4.13.3.1 par_QR	28
4.14	src/Rcc	ommand_Sreport.c File Reference	28
	4.14.1	Detailed Description	29
	4.14.2	Variable Documentation	29
		4.14.2.1 par_SR	29
4.15	src/Sre	port.c File Reference	29
	4.15.1	Detailed Description	30
	4.15.2	Variable Documentation	30
		4.15.2.1 par_SR	30
4.16	src/sta	ts_info.c File Reference	30
	4.16.1	Detailed Description	32
	4.16.2	Function Documentation	32
		4.16.2.1 get_tile_lane	32
		4.16.2.2 init_info	32
		4.16.2.3 resize_info	32
		4.16.2.4 update_ACGT_counts	32
	4.16.3	Variable Documentation	33
		4.16.3.1 par_QR	33
4.17	src/str_	_manip.c File Reference	33
	4.17.1	Detailed Description	33
Index			34

# **Chapter 1**

# **Class Index**

## 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_fq_read		
	Stores a fastq entry	5
_iparam_	_Qreport	
	Qreport input parameters	5
_iparam_	Sreport	
	Sreport input parameters	7
statsinfo		
	Stores info needed to create the summary graphs	7

2 Class Index

# **Chapter 2**

# File Index

## 2.1 File List

Here is a list of all documented files with brief descriptions:

include/defines.n	
Macro definitions	11
include/fopen_gen.h	
Uncompress/compress input/output files using pipes	12
include/fq_read.h	
Fastq entries manipulations (read/write)	14
include/init_Qreport.h	
Header file: help dialog for Qreport and initialization of the command line arguments	16
include/init_Sreport.h	??
include/Rcommand_Qreport.h	
Get Rscript command for Qreport	17
include/Rcommand_Sreport.h	
Get Rscript command for Sreport	18
include/stats_info.h	
Construct the quality report variables and update them	18
include/str_manip.h	
Functions that do string manipulation	21
src/fopen_gen.c	
Uncompress/compress input/output files using pipes	21
src/fq_read.c	
Fastq entries manipulations (read/write)	23
src/init_Qreport.c	
Help dialog for Qreport and initialization of the command line arguments	25
src/init_Sreport.c	
Help dialog for Sreport and initialization of the command line arguments	26
src/Qreport.c	
QReport main function	27
src/Rcommand_Sreport.c	
Get Rscript command for Sreport	28
src/Sreport.c	
Sreport main function	29
src/stats_info.c	
Construct the quality report variables and update them	30
src/str_manip.c	
Functions that do string manipulation	33

File Index

## **Chapter 3**

## **Class Documentation**

## 3.1 \_fq\_read Struct Reference

#### stores a fastq entry

```
#include <fq_read.h>
```

#### **Public Attributes**

- char line1 [READ\_MAXLEN]
- · char line2 [READ MAXLEN]
- char line3 [READ\_MAXLEN]
- char line4 [READ\_MAXLEN]
- int L
- int start

## 3.1.1 Detailed Description

stores a fastq entry

#### 3.1.2 Member Data Documentation

```
3.1.2.1 int _fq_read::L
```

read length

3.1.2.2 int \_fq\_read::start

nucleotide position start. Can only be different from zero if the read has been filtered with this tool.

The documentation for this struct was generated from the following file:

• include/fq\_read.h

## 3.2 \_iparam\_Qreport Struct Reference

#### contains Qreport input parameters

```
#include <init_Qreport.h>
```

6 Class Documentation

#### **Public Attributes**

- char \* inputfile
- char outputfilebin [MAX\_FILENAME]
- char outputfilehtml [MAX\_FILENAME]
- char outputfileinfo [MAX\_FILENAME]
- int nQ
- int ntiles
- int minQ
- int read len
- int filter
- int one\_read\_len

#### 3.2.1 Detailed Description

contains Qreport input parameters

#### 3.2.2 Member Data Documentation

3.2.2.1 int \_iparam\_Qreport::filter

0 original data, 1 this tool filtered data, 2 other tool filtered data

3.2.2.2 char\* \_iparam\_Qreport::inputfile

Inputfile name

3.2.2.3 int \_iparam\_Qreport::minQ

minimum Quality allowed 0 - 45

3.2.2.4 int \_iparam\_Qreport::nQ

# different quality values (default is 46)

3.2.2.5 int \_iparam\_Qreport::ntiles

# tiles (default is 96)

3.2.2.6 int \_iparam\_Qreport::one\_read\_len

1 all reads of equal length 0 reads have different lengths.

3.2.2.7 char \_iparam\_Qreport::outputfilebin[MAX\_FILENAME]

Binary outputfile name.

3.2.2.8 char \_iparam\_Qreport::outputfilehtml[MAX\_FILENAME]

html outputfile name

3.2.2.9 char \_iparam\_Qreport::outputfileinfo[MAX\_FILENAME]

Info outputfile name

3.2.2.10 int \_iparam\_Qreport::read\_len

original read length

The documentation for this struct was generated from the following file:

• include/init Qreport.h

## 3.3 \_iparam\_Sreport Struct Reference

contains Sreport input parameters

```
#include <init_Sreport.h>
```

#### **Public Attributes**

- · char \* inputfolder
- char outputfile [MAX\_FILENAME]

#### 3.3.1 Detailed Description

contains Sreport input parameters

#### 3.3.2 Member Data Documentation

3.3.2.1 char\* \_iparam\_Sreport::inputfolder

Outputfile name

3.3.2.2 char \_iparam\_Sreport::outputfile[MAX\_FILENAME]

html outputfile name

The documentation for this struct was generated from the following file:

· include/init\_Sreport.h

#### 3.4 statsinfo Struct Reference

stores info needed to create the summary graphs

```
#include <stats_info.h>
```

#### **Public Attributes**

- int read len
- int ntiles

8 Class Documentation

- int nQ
- int minQ
- int tile\_pos
- · int nreads
- int reads\_wN
- int sz\_lowQ\_ACGT\_tile
- int sz\_ACGT\_tile
- int sz\_reads\_MlowQ
- int sz\_QPosTile\_table
- int sz\_ACGT\_pos
- int \* tile\_tags
- int \* lane\_tags
- int \* qual\_tags
- uint64\_t \* lowQ\_ACGT\_tile
- uint64\_t \* ACGT\_tile
- uint64\_t \* reads\_MlowQ
- uint64\_t \* QPosTile\_table
- uint64\_t \* ACGT\_pos

#### 3.4.1 Detailed Description

stores info needed to create the summary graphs

- 3.4.2 Member Data Documentation
- 3.4.2.1 uint64\_t\* statsinfo::ACGT\_pos
- # A, C, G, T, N per position
- 3.4.2.2 uint64\_t\* statsinfo::ACGT\_tile
- # A, C, G, T, N per tile, to compute the fraction of lowQuality bases per tile and per nucleotide.
- 3.4.2.3 int\* statsinfo::lane\_tags

Names of the existing tiles

- 3.4.2.4 uint64\_t\* statsinfo::lowQ\_ACGT\_tile
- # low Quality A, C, G, T, N per tile
- 3.4.2.5 int statsinfo::minQ

Minimum quality threshold

- 3.4.2.6 int statsinfo::nQ
- # possible quality values

3.4.2.7 int statsinfo::nreads

# reads read till current position.

3.4.2.8 int statsinfo::ntiles

# tiles

3.4.2.9 uint64\_t\* statsinfo::QPosTile\_table

# bases of a given quality per tile.

3.4.2.10 int\* statsinfo::qual\_tags

Names of the existing qualities

3.4.2.11 int statsinfo::read\_len

Maximum length of a read

3.4.2.12 uint64\_t\* statsinfo::reads\_MlowQ

# reads with M(position) lowQuality bases.

3.4.2.13 int statsinfo::reads\_wN

# reads with N's found till current position

3.4.2.14 int statsinfo::sz\_ACGT\_pos

ACGT\_pos size = read\_len \* N\_ACGT

3.4.2.15 int statsinfo::sz\_ACGT\_tile

ACGT tile size = ntiles \* NACGT

3.4.2.16 int statsinfo::sz\_lowQ\_ACGT\_tile

lowQ\_ACGT\_tile size = ntiles \* N\_ACGT

3.4.2.17 int statsinfo::sz\_QPosTile\_table

QposTile\_Table size = ntiles \* nQ \* read\_len

3.4.2.18 int statsinfo::sz\_reads\_MlowQ

reads\_MlowQ size = read\_len + 1

10 Class Documentation

3.4.2.19 int statsinfo::tile\_pos

current tile position

3.4.2.20 int\* statsinfo::tile\_tags

Names of the existing tiles

The documentation for this struct was generated from the following file:

• include/stats\_info.h

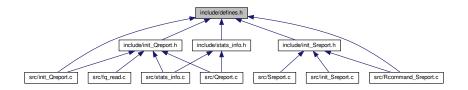
## **Chapter 4**

## **File Documentation**

## 4.1 include/defines.h File Reference

Macro definitions.

This graph shows which files directly or indirectly include this file:



#### Macros

- #define MAX\_FILENAME 300
- #define DEFAULT\_MINQ 27
- #define DEFAULT\_NTILES 96
- #define DEFAULT\_NQ 46
- #define ZEROQ 33
- #define N\_ACGT 5
- #define MAX\_RCOMMAND 4000

#### 4.1.1 Detailed Description

Macro definitions.

**Author** 

Paula Perez paulaperezrubio@gmail.com

Date

07.08.2017

#### 4.1.2 Macro Definition Documentation

4.1.2.1 #define DEFAULT\_MINQ 27

Minimum quality threshold

4.1.2.2 #define DEFAULT\_NQ 46

Default number of different quality values

4.1.2.3 #define DEFAULT\_NTILES 96

Default number of tiles

4.1.2.4 #define MAX\_FILENAME 300

Maximum # chars in a filename

4.1.2.5 #define MAX\_RCOMMAND 4000

Maximum # chars in R command

4.1.2.6 #define N\_ACGT 5

Number of different nucleotides in the fq file

4.1.2.7 #define ZEROQ 33

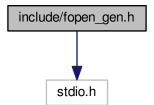
ASCII code of lowest quality value (!)

## 4.2 include/fopen\_gen.h File Reference

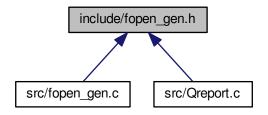
Uncompress/compress input/output files using pipes.

#include <stdio.h>

Include dependency graph for fopen\_gen.h:



This graph shows which files directly or indirectly include this file:



#### **Macros**

- #define **READ\_END** 0
- #define WRITE END 1
- #define PERMISSIONS 0640

#### **Functions**

- int setCloexec (int fd)
- FILE \* fopen\_gen (const char \*path, const char \*mode)

Generalized fopen function. fopen\_gen is to be used as fopen. Can be used in read and in write mode. When used in read mode with a compressed extension, the file will be first decompressed and then read. When used in write mode with a compressed extension, the output will be compressed.

#### 4.2.1 Detailed Description

Uncompress/compress input/output files using pipes.

Hook the standard file opening functions, open, fopen and fopen64. If the extension of the file being opened indicates the file is compressed (.gz, .bz2, .xz), when opening in the reading mode a pipe to a program is opened that decompresses that file (gunzip, bunzip2 or xzdec) and return a handle to the open pipe. When opening in the writing mode (only for .gz, .bam), a pipe to a program is opened that compresses the output.

#### **Author**

Paula Perez paulaperez rubio@gmail.com

Date

03.08.2017

#### Warning

vfork vs fork to be checked!

#### Note

- original copyright note - (reading mode, original C++ code) author: Shaun Jackman sjackman@bcgsc. ca, https://github.com/bcgsc, filename: Uncompress.cpp

#### 4.2.2 Function Documentation

4.2.2.1 FILE\* fopen\_gen ( const char \* path, const char \* mode )

Generalized fopen function. fopen\_gen is to be used as fopen. Can be used in read and in write mode. When used in read mode with a compressed extension, the file will be first decompressed and then read. When used in write mode with a compressed extension, the output will be compressed.

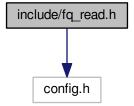
Returns

a FILE pointer

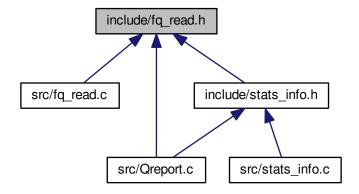
## 4.3 include/fq\_read.h File Reference

fastq entries manipulations (read/write)

#include "config.h"
Include dependency graph for fq\_read.h:



This graph shows which files directly or indirectly include this file:



#### Classes

struct \_fq\_read
 stores a fastq entry

## **Typedefs**

 typedef struct \_fq\_read Fq\_read stores a fastq entry

#### **Functions**

```
    void get_fqread (Fq_read *seq, char *buffer, int c1, int c2, int k)
    reads fastq line from a buffer
```

int string\_seq (Fq\_read \*seq, char \*char\_seq)
 writes the fq entry in a string

#### 4.3.1 Detailed Description

fastq entries manipulations (read/write)

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

03.08.2017

#### 4.3.2 Function Documentation

4.3.2.1 void get\_fqread ( Fq\_read \* seq, char \* buffer, int pos1, int pos2, int nline )

reads fastq line from a buffer

a fastq line is read from a buffer and the relevant information is stored in a structure **Fq\_read**. Depending on the variable **par\_QR** values, information about whether the read was trimmed is stored.

#### **Parameters**

*seq	pointer to Fq_read, where the info will be stored.
buffer	variable where the file being read is stored.
pos1	buffer start position of the line.
pos2	buffer end position of the line.
nline	file line number being read.

4.3.2.2 int string\_seq ( Fq\_read \* seq, char \* char\_seq )

writes the fq entry in a string

#### **Parameters**

*seq	pointer to <b>Fq_read</b> , where the info will be stored.
char_seq	pointer to buffer, where the sequence will be stored

#### Warning

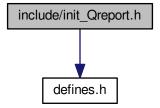
change the call to sprintf to snprintf

## 4.4 include/init\_Qreport.h File Reference

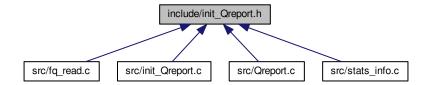
Header file: help dialog for Qreport and initialization of the command line arguments.

#include "defines.h"

Include dependency graph for init\_Qreport.h:



This graph shows which files directly or indirectly include this file:



#### Classes

struct \_iparam\_Qreport
 contains Qreport input parameters

#### **Typedefs**

 typedef struct \_iparam\_Qreport Iparam\_Qreport contains Qreport input parameters

#### **Functions**

void printHelpDialog\_Qreport ()

Function that prints Qreport help dialog when called.

• void getarg\_Qreport (int argc, char \*\*argv)

Reads in the arguments passed through the command line to Qreport. and stores them in the global variable par\_QR.

#### 4.4.1 Detailed Description

Header file: help dialog for Qreport and initialization of the command line arguments.

**Author** 

Paula Perez paulaperezrubio@gmail.com

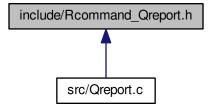
Date

03.08.2017

## 4.5 include/Rcommand\_Qreport.h File Reference

get Rscript command for Qreport

This graph shows which files directly or indirectly include this file:



#### **Functions**

• char \* command\_Qreport ()

returns Rscript command that generates the quality report in html

#### 4.5.1 Detailed Description

get Rscript command for Qreport

**Author** 

Paula Perez paulaperezrubio@gmail.com

Date

07.08.2017

**Author** 

Paula Perez paulaperez rubio@gmail.com

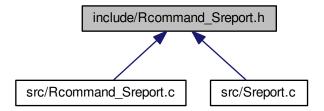
Date

09.08.2017

## 4.6 include/Rcommand\_Sreport.h File Reference

get Rscript command for Sreport

This graph shows which files directly or indirectly include this file:



#### **Functions**

char \* command\_Sreport ()
 returns Rscript command that generates the summary report in html

#### 4.6.1 Detailed Description

get Rscript command for Sreport

Author

Paula Perez paulaperez rubio@gmail.com

Date

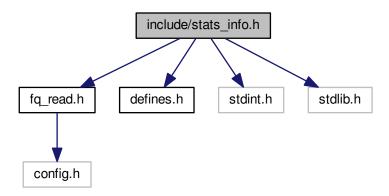
09.08.2017

## 4.7 include/stats\_info.h File Reference

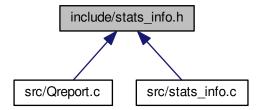
Construct the quality report variables and update them.

```
#include "fq_read.h"
#include "defines.h"
#include <stdint.h>
#include <stdlib.h>
```

Include dependency graph for stats\_info.h:



This graph shows which files directly or indirectly include this file:



#### **Classes**

struct statsinfo

stores info needed to create the summary graphs

## **Typedefs**

 typedef struct statsinfo Info stores info needed to create the summary graphs

#### **Functions**

void init\_info (Info \*res)

```
Initialization of a Info type.

    void free_info (Info *res)

      frees allocated memory in Info

    void read info (Info *res, char *file)

      Read Info from binary file.

    void write_info (Info *res, char *file)

      Write info to binary file.

    void print info (Info *res, char *infofile)

      print Info to a textfile

    void get_first_tile (Info *res, Fq_read *seq)

      gets first tile
void update_info (Info *res, Fq_read *seq)
      updates Info with Fq read

    int update_ACGT_counts (uint64_t *ACGT_low, char ACGT)

      update, for current tile, ACGT counts.

    void update_QPosTile_table (Info *res, Fq_read *seq)

      update QPostile table

    void update_ACGT_pos (uint64_t *ACGT_pos, Fq_read *seq, int read_len)

      update ACGT_pos
void resize_info (Info *res)
      resize Info
```

#### 4.7.1 Detailed Description

Construct the quality report variables and update them.

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

04.08.2017

#### 4.7.2 Function Documentation

```
4.7.2.1 void init_info ( Info * res )
```

Initialization of a Info type.

It sets: nQ, read len, ntiles, minQ and the dimensions of the arrays. Initializes the rest of the variables to zero and allocates memory to the arrays initializing them to 0 (calloc).

```
4.7.2.2 void resize_info ( Info * res )
```

resize Info

At the end of the program, resize the structure Info, and adapt it to the actual number of tiles and the actual number of different quality values present.

```
4.7.2.3 int update_ACGT_counts ( uint64_t * ACGT_low, char ACGT )
```

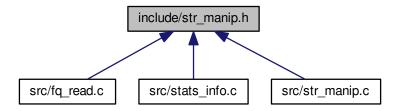
update, for current tile, ACGT counts.

Makes update of ACGT counts for the current tile. Can be used with variables: lowQ\_ACGT\_tile and ACGT\_tile

## 4.8 include/str\_manip.h File Reference

functions that do string manipulation

This graph shows which files directly or indirectly include this file:



#### **Functions**

```
    int strindex (char *s, char *t)
    returns index of t in s (start, first occurence)
```

• int count\_char (char \*s, char c)

returns the # of occurences of char c in string s

#### 4.8.1 Detailed Description

functions that do string manipulation

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

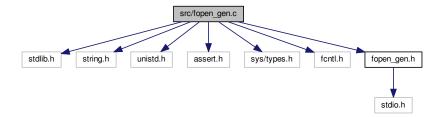
03.08.2017

## 4.9 src/fopen\_gen.c File Reference

Uncompress/compress input/output files using pipes.

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <assert.h>
#include <sys/types.h>
#include <fcntl.h>
#include "fopen_gen.h"
```

Include dependency graph for fopen\_gen.c:



#### **Functions**

- static const char \* zcatExec (const char \*path)
- static const char \* catExec (const char \*path)

Commands to compress files. To be done in output.

static int uncompress (const char \*path)

Open a pipe to uncompress file. Open a pipe to uncompress the specified file. Not thread safe.

static int compress (const char \*path)

Open a pipe to compress output. Open a pipe to uncompress the specified file. Not thread safe.

- int setCloexec (int fd)
- static FILE \* funcompress (const char \*path)

Open a pipe to uncompress the specified file.

static FILE \* fcompress (const char \*path)

Open a pipe to compress the specified file.

FILE \* fopen\_gen (const char \*path, const char \*mode)

Generalized fopen function. fopen\_gen is to be used as fopen. Can be used in read and in write mode. When used in read mode with a compressed extension, the file will be first decompressed and then read. When used in write mode with a compressed extension, the output will be compressed.

#### 4.9.1 Detailed Description

Uncompress/compress input/output files using pipes.

Hook the standard file opening functions, open, fopen and fopen64. If the extension of the file being opened indicates the file is compressed (.gz, .bz2, .xz), when opening in the reading mode a pipe to a program is opened that decompresses that file (gunzip, bunzip2 or xzdec) and return a handle to the open pipe. When opening in the writing mode (only for .gz, .bam), a pipe to a program is opened that compresses the output.

**Author** 

Paula Perez paulaperezrubio@gmail.com

Date

03.08.2017

Warning

vfork vs fork to be checked!

Note

- original copyright note - (reading mode, original C++ code) author: Shaun Jackman sjackman@bcgsc. ca, https://github.com/bcgsc, filename: Uncompress.cpp

#### 4.9.2 Function Documentation

```
4.9.2.1 static int compress (const char * path ) [static]
```

Open a pipe to compress output. Open a pipe to uncompress the specified file. Not thread safe.

Returns

a file descriptor

```
4.9.2.2 static FILE* fcompress ( const char * path ) [static]
```

Open a pipe to compress the specified file.

Returns

a FILE pointer

```
4.9.2.3 FILE* fopen_gen ( const char * path, const char * mode )
```

Generalized fopen function. fopen\_gen is to be used as fopen. Can be used in read and in write mode. When used in read mode with a compressed extension, the file will be first decompressed and then read. When used in write mode with a compressed extension, the output will be compressed.

Returns

a FILE pointer

```
4.9.2.4 static FILE* funcompress (const char * path ) [static]
```

Open a pipe to uncompress the specified file.

Returns

a FILE pointer

```
4.9.2.5 static int uncompress (const char * path ) [static]
```

Open a pipe to uncompress file. Open a pipe to uncompress the specified file. Not thread safe.

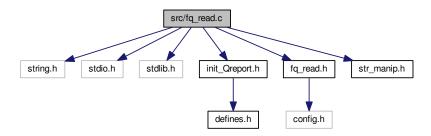
Returns

a file descriptor

## 4.10 src/fq\_read.c File Reference

fastq entries manipulations (read/write)

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "init_Qreport.h"
#include "fq_read.h"
#include "str_manip.h"
Include dependency graph for fq_read.c:
```



#### **Functions**

- void get\_fqread (Fq\_read \*seq, char \*buffer, int pos1, int pos2, int nline)
   reads fastq line from a buffer
- int string\_seq (Fq\_read \*seq, char \*char\_seq)
   writes the fq entry in a string

#### **Variables**

• Iparam\_Qreport par\_QR

#### 4.10.1 Detailed Description

fastq entries manipulations (read/write)

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

03.08.2017

#### 4.10.2 Function Documentation

```
4.10.2.1 void get_fqread ( Fq_read * seq, char * buffer, int pos1, int pos2, int nline )
```

reads fastq line from a buffer

a fastq line is read from a buffer and the relevant information is stored in a structure **Fq\_read**. Depending on the variable **par\_QR** values, information about whether the read was trimmed is stored.

#### **Parameters**

*seq	pointer to <b>Fq_read</b> , where the info will be stored.
buffer	variable where the file being read is stored.
pos1	buffer start position of the line.
pos2	buffer end position of the line.
nline	file line number being read.

#### 4.10.2.2 int string\_seq ( Fq\_read \* seq, char \* char\_seq )

writes the fq entry in a string

#### **Parameters**

*seq	pointer to Fq_read, where the info will be stored.
char_seq	pointer to buffer, where the sequence will be stored

#### Warning

change the call to sprintf to snprintf

#### 4.10.3 Variable Documentation

#### 4.10.3.1 Iparam\_Qreport par\_QR

input parameters

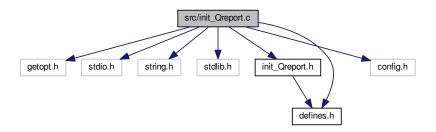
global variable: input parameters

## 4.11 src/init\_Qreport.c File Reference

Help dialog for Qreport and initialization of the command line arguments.

```
#include <getopt.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "init_Qreport.h"
#include "config.h"
#include "defines.h"
```

Include dependency graph for init\_Qreport.c:



#### **Functions**

• void printHelpDialog\_Qreport ()

Function that prints Qreport help dialog when called.

void getarg\_Qreport (int argc, char \*\*argv)

Reads in the arguments passed through the command line to Qreport. and stores them in the global variable par\_QR.

#### **Variables**

· Iparam Qreport par QR

#### 4.11.1 Detailed Description

Help dialog for Qreport and initialization of the command line arguments.

**Author** 

```
Paula Perez paulaperez rubio@gmail.com
```

Date

03.08.2017

#### 4.11.2 Variable Documentation

#### 4.11.2.1 Iparam\_Qreport par\_QR

input parameters

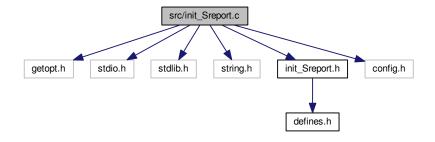
global variable: input parameters

## 4.12 src/init\_Sreport.c File Reference

Help dialog for Sreport and initialization of the command line arguments.

```
#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "init_Sreport.h"
#include "config.h"
```

Include dependency graph for init\_Sreport.c:



#### **Functions**

• void printHelpDialog\_Sreport ()

Function that prints Sreport help dialog when called.

void getarg\_Sreport (int argc, char \*\*argv)

Reads in the arguments passed through the command line to Sreport. and stores them in the global variable par SR.

#### **Variables**

· Iparam Sreport par SR

#### 4.12.1 Detailed Description

Help dialog for Sreport and initialization of the command line arguments.

**Author** 

```
Paula Perez paulaperez rubio@gmail.com
```

Date

09.08.2017

#### 4.12.2 Variable Documentation

#### 4.12.2.1 Iparam\_Sreport par\_SR

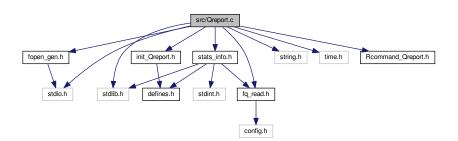
input parameters Sreport

## 4.13 src/Qreport.c File Reference

#### QReport main function.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "init_Qreport.h"
#include "fopen_gen.h"
#include "fq_read.h"
#include "stats_info.h"
#include "Rcommand_Qreport.h"
```

Include dependency graph for Qreport.c:



#### **Macros**

• #define B\_LEN 131072

#### **Functions**

int main (int argc, char \*argv[])
 Qreport main function.

#### Variables

Iparam\_Qreport par\_QR

#### 4.13.1 Detailed Description

QReport main function.

**Author** 

Paula Perez paulaperezrubio@gmail.com

#### Date

03.08.2017 This file contains the quality report main function. It reads a fastq file and creates a html quality report. See README\_Qreport.md for more details.

#### 4.13.2 Macro Definition Documentation

```
4.13.2.1 #define B_LEN 131072
```

buffer size

#### 4.13.3 Variable Documentation

```
4.13.3.1 Iparam_Qreport par_QR
```

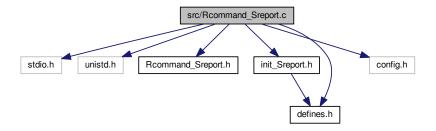
global variable: input parameters

## 4.14 src/Rcommand\_Sreport.c File Reference

#### get Rscript command for Sreport

```
#include <stdio.h>
#include <unistd.h>
#include "Rcommand_Sreport.h"
#include "init_Sreport.h"
#include "defines.h"
#include "config.h"
```

Include dependency graph for Rcommand\_Sreport.c:



#### **Functions**

char \* command\_Sreport ()
 returns Rscript command that generates the summary report in html

#### **Variables**

• Iparam\_Sreport par\_SR

#### 4.14.1 Detailed Description

get Rscript command for Sreport

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

09.08.2017

#### 4.14.2 Variable Documentation

4.14.2.1 Iparam\_Sreport par\_SR

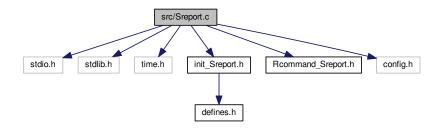
input parameters Sreport

## 4.15 src/Sreport.c File Reference

#### Sreport main function.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "init_Sreport.h"
#include "Rcommand_Sreport.h"
#include "config.h"
```

Include dependency graph for Sreport.c:



## **Functions**

int main (int argc, char \*argv[])
 Qreport main function.

#### **Variables**

Iparam\_Sreport par\_SR

#### 4.15.1 Detailed Description

Sreport main function.

**Author** 

Paula Perez paulaperez rubio@gmail.com

Date

09.08.2017 This file contains the summary report main function. Given a folder containing \*bin as from Qreport output, Sreport generates a summary report in html format. See README\_Sreport.md for more details.

#### 4.15.2 Variable Documentation

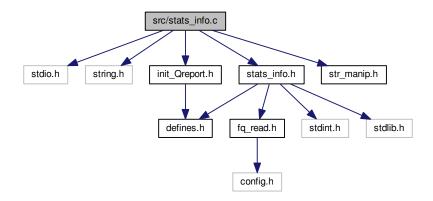
input parameters Sreport

## 4.16 src/stats\_info.c File Reference

Construct the quality report variables and update them.

```
#include <stdio.h>
#include <string.h>
#include "stats_info.h"
#include "init_Qreport.h"
#include "str_manip.h"
```

Include dependency graph for stats\_info.c:



#### **Functions**

```
    void get_tile_lane (char *line1, int *tile, int *lane)
    get tile number from first line in fastq entry.
```

static int belongsto (int k, int \*qual\_tags, int nQ)

returns 1 if k is in qual\_tags, 0 otherwise.

static int cmpfunc (const void \*a, const void \*b)

comparison function for qsort

void init\_info (Info \*res)

Initialization of a Info type.

void free\_info (Info \*res)

frees allocated memory in Info

void read\_info (Info \*res, char \*file)

Read Info from binary file.

void write\_info (Info \*res, char \*file)

Write info to binary file.

• void print\_info (Info \*res, char \*infofile)

print Info to a textfile

void get\_first\_tile (Info \*res, Fq\_read \*seq)

gets first tile

void update info (Info \*res, Fq read \*seq)

updates Info with Fq\_read

• int update\_ACGT\_counts (uint64\_t \*ACGT\_low, char ACGT)

update, for current tile, ACGT counts.

void update\_QPosTile\_table (Info \*res, Fq\_read \*seq)

update QPostile table

• void update\_ACGT\_pos (uint64\_t \*ACGT\_pos, Fq\_read \*seq, int read\_len)

update ACGT\_pos

void resize\_info (Info \*res)

resize Info

#### **Variables**

Iparam\_Qreport par\_QR

#### 4.16.1 Detailed Description

Construct the quality report variables and update them.

Author

Paula Perez paulaperez rubio@gmail.com

Date

04.08.2017

#### 4.16.2 Function Documentation

```
4.16.2.1 void get_tile_lane ( char * line1, int * tile, int * lane )
```

get tile number from first line in fastq entry.

#### **Parameters**

line1	first line of a fastq entry
tile	int∗ where the tile will be stored
lane	int∗ where the lane will be stored

#### See also

```
http://wiki.christophchamp.com/index.php?title=FASTQ_format
```

Only Illumina sequence identifiers are allowed. The line is inspected, and the number of ':' is obtained. The function exits with an error if the number of semicolons is different from 4 or 9.

```
4.16.2.2 void init_info ( Info * res )
```

Initialization of a Info type.

It sets: nQ, read\_len, ntiles, minQ and the dimensions of the arrays. Initializes the rest of the variables to zero and allocates memory to the arrays initializing them to 0 (calloc).

```
4.16.2.3 void resize_info ( Info * res )
```

resize Info

At the end of the program, resize the structure Info, and adapt it to the actual number of tiles and the actual number of different quality values present.

```
4.16.2.4 int update_ACGT_counts ( uint64_t * ACGT_low, char ACGT )
```

update, for current tile, ACGT counts.

Makes update of ACGT counts for the current tile. Can be used with variables: lowQ\_ACGT\_tile and ACGT\_tile

#### 4.16.3 Variable Documentation

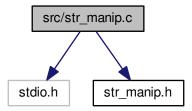
#### 4.16.3.1 Iparam\_Qreport par\_QR

global variable: input parameters

## 4.17 src/str\_manip.c File Reference

functions that do string manipulation

```
#include <stdio.h>
#include "str_manip.h"
Include dependency graph for str_manip.c:
```



#### **Functions**

- int strindex (char \*s, char \*t)
   returns index of t in s (start, first occurence)
- int count\_char (char \*s, char c)

  returns the # of occurences of char c in string s

#### 4.17.1 Detailed Description

functions that do string manipulation

**Author** 

Paula Perez paulaperezrubio@gmail.com

Date

03.08.2017

# Index

```
nreads
statsinfo, 8
ntiles
statsinfo, 9
statsinfo, 7
nreads, 8
ntiles, 9
```