
proto_err Documentation

Release 1

Phelim Bradley

December 09, 2013

CONTENTS

1	proto_err – Main package	1
1.1	proto_err.simulation – Simulating Error Reads	1
1.2	proto_err.errorCount – Error Counting	1
2	Indices and tables	5
	Python Module Index	7
	Index	9

PROTO_ERR – MAIN PACKAGE

1.1 `proto_err.simulation` – Simulating Error Reads

```
class proto_err.simulation.complexError(record, opt, id)
    Bases: proto_err.simulation.simulateError

    error()
        Function to induce and error

class proto_err.simulation.simulateError(record, opt, id)
    Class of objects to simulate errors in a SeqRecord

    deletion(pos, dlen)
        function to induce a deletion

    indel(pos)
        function to induce an INDEL

    ins(pos, rl)
        function to induce a insertion

    qscore None

    record None
        Return a Bio.SeqRecord

    snp(pos, rl)
        function to induce a SNP

class proto_err.simulation.singleSNP(record, opt, id)
    Bases: proto_err.simulation.simulateError

    error()
        Function to induce and error

proto_err.simulation.subsample(ref, opt, errorSimulator=<class
    proto_err.simulation.complexError at 0x10540fc18>)
    Function to take a fasta file subsample reads and generate a list of subsampled reads
```

1.2 `proto_err.errorCount` – Error Counting

```
class proto_err.errorCount.counter(ref, errorList=None, samfile=None, opt=None)
    Takes a list of errors and does some kmer counting

    countAlignedBases(read)
```

countErrorKmer (*maxKmerLength=None*)

Function which takes a list of errors and counts the kmers before and after and error

countRefKmer (*maxKmerLength=None*)

Function to count all kmers in long sequence (reference) of length kmerLen or below

getCount (*truth=None, emission=None, kmer=' ', after=False*)

Gets the count for a given {truth,emmission,kmer}

kmerFreq (*seq, kmer*)

Function to calculate the number of times a kmer appears in a sequence

readDiff (*read, ref*)

setup (*opt*)

Function to set up dictionary structure for outputed stats

class `proto_err.errorCount.error` (*true, emission, read, readPos*)

Information about the errors in a read

after (*j*)

Return the following j bases,return N when bases missing

before (*j*)

Return the preceding j bases,return N when bases missing

emissionSeq `None`

Sequence emmited

errorType `None`

isDeletion `None`

Is the error a deletion

isIndel `None`

Is the error an INDEL

isInsertion `None`

Is the error an insertion

isSnp `None`

Is the error a SNP

qscore (*i*)

Return the quality score at a base +i i from the error start position

qual `None`

Quality score of error base

trueSeq `None`

Sequence of the truth

class `proto_err.errorCount.errorReader` (*samfile, ref*)

Iterable over errors in aligned reads

checkDeletion (*N*)

checkHardClipped (*N*)

checkInsertion (*N*)

checkPadding (*N*)

checkRead ()

A function which take a read and generates some error objects

checkSNPs (*N*)

checkSeqMismatch (*N*)

checkSkipped (*N*)

checkSoftClipped (*N*)

next ()

readNext ()

refRead **None**

Function to return the reference sequence a read is aligned to

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

p

`proto_err`, 1
`proto_err.errorCount`, 1
`proto_err.simulation`, 1

A

after() (proto_err.errorCount.error method), 2

B

before() (proto_err.errorCount.error method), 2

C

checkDeletion() (proto_err.errorCount.errorReader method), 2

checkHardClipped() (proto_err.errorCount.errorReader method), 2

checkInsertion() (proto_err.errorCount.errorReader method), 2

checkPadding() (proto_err.errorCount.errorReader method), 2

checkRead() (proto_err.errorCount.errorReader method), 2

checkSeqMismatch() (proto_err.errorCount.errorReader method), 3

checkSkipped() (proto_err.errorCount.errorReader method), 3

checkSNPs() (proto_err.errorCount.errorReader method), 2

checkSoftClipped() (proto_err.errorCount.errorReader method), 3

complexError (class in proto_err.simulation), 1

countAlignedBases() (proto_err.errorCount.counter method), 1

counter (class in proto_err.errorCount), 1

countErrorKmer() (proto_err.errorCount.counter method), 1

countRefKmer() (proto_err.errorCount.counter method), 2

D

deletion() (proto_err.simulation.simulateError method), 1

E

emissionSeq (proto_err.errorCount.error attribute), 2

error (class in proto_err.errorCount), 2

error() (proto_err.simulation.complexError method), 1

error() (proto_err.simulation.singleSNP method), 1

errorReader (class in proto_err.errorCount), 2

errorType (proto_err.errorCount.error attribute), 2

G

getCount() (proto_err.errorCount.counter method), 2

I

indel() (proto_err.simulation.simulateError method), 1

ins() (proto_err.simulation.simulateError method), 1

isDeletion (proto_err.errorCount.error attribute), 2

isIndel (proto_err.errorCount.error attribute), 2

isInsertion (proto_err.errorCount.error attribute), 2

isSnp (proto_err.errorCount.error attribute), 2

K

kmerFreq() (proto_err.errorCount.counter method), 2

N

next() (proto_err.errorCount.errorReader method), 3

P

proto_err (module), 1

proto_err.errorCount (module), 1

proto_err.simulation (module), 1

Q

qscore (proto_err.simulation.simulateError attribute), 1

qscore() (proto_err.errorCount.error method), 2

qual (proto_err.errorCount.error attribute), 2

R

readDiff() (proto_err.errorCount.counter method), 2

readNext() (proto_err.errorCount.errorReader method), 3

record (proto_err.simulation.simulateError attribute), 1

refRead (proto_err.errorCount.errorReader attribute), 3

S

setup() (proto_err.errorCount.counter method), 2

simulateError (class in proto_err.simulation), 1

singleSNP (class in proto_err.simulation), 1

snp() (proto_err.simulation.simulateError method), 1

`subsample()` (in module `proto_err.simulation`), 1

T

`trueSeq` (`proto_err.errorCount.error` attribute), 2