proto_err Documentation

Release 1

Phelim Bradley

December 09, 2013

CONTENTS

Contents:

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

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,emmision,kmer}

kmerFreq(*seq*, *kmer*)
Function to calculate the number of times a kmer appears in a sequence

setup(*opt*)
Function to set up dictionary structure for outputed stats

class 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

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 errorCount.**errorReader**(*samfile*, *ref*)
Iterable over errors in aligned reads

checkRead()
A function which take a read and generates some error objects

refRead None
Function to return the reference sequence a read is aligned to

class 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

record **None**
Return a Bio.SeqRecord

snp (*pos*, *rl*)
function to induce a SNP

simulation.**subsample** (*ref*, *opt*, *errorSimulator=<class simulation.complexError at 0x106fc6188>*)
Function to take a fasta file subsample reads and generate a list of subsampled reads

fastaIO.**writeFasta** (*filename*, *seqList*)
Function to take list of Seq objects and write a fasta file

fastaIO.**writeFastq** (*filename*, *seqList*)
Function to take list of Seq objects and write a fasta file

class align.**BwaMemAlignCommandline** (*cmd='bwa'*, ***kwargs*)
Command line wrapper for Burrows Wheeler Aligner (BWA) aln.

Run a BWA alignment, equivalent to:

```
$ bwa aln [...] <in.db.fasta> <in.query.fq> > <out.sai>
```

See http://bio-bwa.sourceforge.net/bwa.shtml for details.

Example:

```
>>> from Bio.Sequencing.Applications import BwaAlignCommandline
>>> reference_genome = "/path/to/reference_genome.fasta"
>>> read_file = "/path/to/read_1.fq"
>>> output_sai_file = "/path/to/read_1.sai"
>>> read_group="@RG ID:foo  SM:bar"
>>> align_cmd = BwaAlignCommandline(reference=reference_genome, read_file=read_file)
>>> print(align_cmd)
bwa aln /path/to/reference_genome.fasta /path/to/read_1.fq
```

You would typically run the command line using align_cmd(stdout=output_sai_file) or via the Python subprocess module, as described in the Biopython tutorial.

align.**refIndex** (*file*)
Function to generate BWA index

CHAPTER

ONE

INDICES AND TABLES

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

a

e

f

s