# seq-tools Documentation

Release 0.2.0

Jason L Weirather

1	seqto	tools package 3					
	1.1	Subpackages	3				
		1.1.1 seqtools.cli package	3				
		Subpackages	3				
		Submodules	5				
		seqtools.cli.cli_front module	5				
		Module contents	6				
		1.1.2 seqtools.format package	6				
		Submodules	6				
		seqtools.format.bamindex module	6				
		seqtools.format.bed module	8				
		seqtools.format.bgzf module	8				
		seqtools.format.fasta module	9				
		*					
		seqtools.format.fastq module	10				
		seqtools.format.gpd module	10				
		seqtools.format.psl module	11				
		seqtools.format.sam module	12				
		Module contents	17				
		1.1.3 seqtools.simulation package	17				
		Submodules	17				
		seqtools.simulation.emitter module	17				
		seqtools.simulation.permute module	17				
		seqtools.simulation.randomsource module	19				
		Module contents	20				
	1.2	Submodules	20				
	1.3	seqtools.align module	20				
	1.4	seqtools.errors module	22				
	1.5	seqtools.graph module	28				
	1.6	seqtools.range module	31				
	1.7	seqtools.sequence module	37				
	1.8	seqtools.statistics module	38				
	1.9	seqtools.stream module	39				
	1.10	segtools.structure module	40				
	1.11	Module contents	49				
	1,11	Tiodate volumes	1)				
2	Indic	ces and tables	51				
Py	Python Module Index						

Index 55

Contents:

Contents 1

2 Contents

# seqtools package

# **Subpackages**

### seqtools.cli package

### **Subpackages**

seqtools.cli.utilities package

#### **Submodules**

**seqtools.cli.utilities.bam\_bgzf\_index module** The bam bgzf utility produces gzipped index of a bam file with the following fields for each line of the bam file (not counting header)

- 1. Query name
- 2. Target range
- 3. BlockStart
- 4. InnerStart
- 5. Aligned Base Count
- 6. Flag

This index can be used to provide random access into a bam file

```
class seqtools.cli.utilities.bam_bgzf_index.Queue (val)
```

```
get()
```

```
seqtools.cli.utilities.bam_bgzf_index.do_chunk(coords, ecount, args)
seqtools.cli.utilities.bam_bgzf_index.do_inputs()
seqtools.cli.utilities.bam_bgzf_index.external_cmd(cmd)
seqtools.cli.utilities.bam_bgzf_index.main(args)
seqtools.cli.utilities.bam_bgzf_index.setup_tempdir(args)
```

**seqtools.cli.utilities.bam\_to\_bed\_depth module** Convert a BAM or a SAM into a bed depth file

The file is a TSV format with the fields

- 1. Chromosome
- 2. Start (0-index)
- 3. End (1-index)
- 4. Read depth

The file is ordered and covers all regions covered by alignments

```
seqtools.cli.utilities.bam_to_bed_depth.do_inputs()
seqtools.cli.utilities.bam_to_bed_depth.do_output(outputs)
seqtools.cli.utilities.bam_to_bed_depth.external_cmd(cmd)
seqtools.cli.utilities.bam_to_bed_depth.get_output(bedarray, z)
seqtools.cli.utilities.bam_to_bed_depth.main(args)
seqtools.cli.utilities.bam_to_bed_depth.setup_tempdir(args)
```

**seqtools.cli.utilities.fasta\_to\_fake\_fastq module** Convert a FASTA file into a FASTQ file. You can designate what to include in the quality score by setting the –ascii paramater (default 'I')

```
seqtools.cli.utilities.fasta_to_fake_fastq.do_inputs()
seqtools.cli.utilities.fasta_to_fake_fastq.external_cmd(cmd)
seqtools.cli.utilities.fasta_to_fake_fastq.main(args)
```

### seqtools.cli.utilities.fasta\_to\_tsv module Convert a FASTA file into a TSV with the fields

- 1. Header
- 2. Sequence

The Header cannot contain tabs, and any linebreaks in the sequence will be lost

```
seqtools.cli.utilities.fasta_to_tsv.do_inputs()
seqtools.cli.utilities.fasta_to_tsv.external_cmd(cmd)
seqtools.cli.utilities.fasta_to_tsv.main(args)
```

### 

```
seqtools.cli.utilities.fastq_to_fasta.do_inputs()
seqtools.cli.utilities.fastq_to_fasta.external_cmd(cmd)
seqtools.cli.utilities.fastq_to_fasta.main(args)
```

### seqtools.cli.utilities.fastq\_to\_tsv module Convert a FASTQ to a TSV with the following fields

- 1. Header (without the '@' prepending)
- 2. Sequence
- 3. Line 3 (still has the '+' sign)

### 4. Quality

```
lines cannot contain tabs
```

```
seqtools.cli.utilities.fastq_to_tsv.do_inputs()
seqtools.cli.utilities.fastq_to_tsv.external_cmd(cmd)
seqtools.cli.utilities.fastq_to_tsv.main(args)
```

seqtools.cli.utilities.trim\_fasta module left or right trim a FASTA file (option ot invert the trim to keep the ends)

```
seqtools.cli.utilities.trim_fasta.do_inputs()
seqtools.cli.utilities.trim_fasta.external_cmd(cmd)
seqtools.cli.utilities.trim_fasta.main(args)
```

**seqtools.cli.utilities.trim\_fastq module** Trim a FASTQ file/stream ends of all entries (option to invert and only keep the ends)

```
seqtools.cli.utilities.trim_fastq.do_inputs()
seqtools.cli.utilities.trim_fastq.external_cmd(cmd)
seqtools.cli.utilities.trim_fastq.main(args)
```

seqtools.cli.utilities.tsv\_to\_fasta module Undo the fasta\_to\_tsv command and put it back in fasta format

```
seqtools.cli.utilities.tsv_to_fasta.do_inputs()
seqtools.cli.utilities.tsv_to_fasta.external_cmd(cmd)
seqtools.cli.utilities.tsv_to_fasta.main(args)
```

seqtools.cli.utilities.tsv\_to\_fastq module Undo the fastq\_to\_tsv command and put it back in fastq format

```
seqtools.cli.utilities.tsv_to_fastq.do_inputs()
seqtools.cli.utilities.tsv_to_fastq.external_cmd(cmd)
seqtools.cli.utilities.tsv_to_fastq.main(args)
```

#### Module contents

#### **Submodules**

### seqtools.cli.cli\_front module

The cli\_front is a the command line utility that is used to list all the accessable command line utilities and to call the command line utility you want to run.

```
seqtools.cli.cli_front.do_args()
seqtools.cli.cli_front.main()
```

#### **Module contents**

### seqtools.format package

#### **Submodules**

### seqtools.format.bamindex module

bamindex class describes a custom index format used by AlignQC

class seqtools.format.bamindex.BAMIndex (index\_file)

Index file is a gzipped TSV file with these fields:

- 1.qname
- 2.target range
- 3.bgzf file block start
- 4.bgzf inner block start
- 5.aligned base count
- 6.flag

### Usage:

name\_to\_num is used to get all the names at random get\_longest\_target\_alignment\_coords\_by\_name is used to get the best random coord hash is import for random access There are some inactive methods because the datastructures they needed were not getting used and were memory intensive. subsequent updates could put them back or even better only use them when the methods requiring them are called the first time This class is actually incredibly bulky for working with a big index > 1M reads. I think some more specific cases may need to be written

**Parametersindex\_file** (string) – filename (of the gzipped index file)

#### check ordered()

True if each chromosome is listed together as a chunk and if the range starts go from smallest to largest otherwise false

**Returns**is it ordered?

Return typebool

#### destroy()

Try to clear memory up by setting values to None

#### get\_coord\_line\_number (coord)

return the one-indexed line number given the coordinates

get\_coords\_by\_name (name)

Warning: not implemented

```
get_index_line(lnum)
```

Take the 1-indexed line number and return its index information

#### get length()

number of indexed entries

### get\_longest\_target\_alignment\_coords\_by\_name (name)

For a name get the best alignment

**Returns**[filebyte,innerbyte] describing the to distance the zipped block start, and the distance within the unzipped block

```
Return typelist
```

```
get_names()
    get all the query names

get_range_start_coord(rng)

Warning: not implemented

get_range_start_line_number(rng)

Warning: not implemented

get_unaligned_lines()
    get the lines that are not aligned

get_unaligned_start_coord()

Warning: not implemented
```

The best index class will read an index file and only provide accessto primary alignment coordinates

```
Parameters
```

•path - bamfile

•index\_file - bam index to write

•verbose (bool) - default False

```
•samtools (bool) – use samtools default False
seqtools.format.bed module
class seqtools.format.bed.Bed12(bed_line)
     Bases: seqtools.structure.Transcript
     Bed format with 9 optional fields
          Parametersbed_line (string) – one line of a bed file
     get bed line()
          get the bed line
     get_line()
          get the bed line
     value (key)
          access the bed line by key
              Parameterskey (string) – which attribute of the bed12
seqtools.format.bgzf module
seqtools.format.bgzf.get_block_bounds(filename)
     Pre block startsstart 0-indexted, end 1-indexted
          Parametersfilename (string) - filename
          Returns0-index start and 1-index end
          Return typearray of arrays with the [start end] of each block
seqtools.format.bgzf.is_bgzf(filename)
     Pre: filename to test if it is a bgzf format
     Post: True or False
          Parametersfilename (string) -
          Returnsif its a bgzf
          Return typebool
class seqtools.format.bgzf.reader(handle, blockStart=None, innerStart=None)
     Methods adapted from biopython's bgzf.py
          (optional) blockStart is the byte start location of a block (optional) innerStart says how far into a
          decompressed bock to start
          Parameters
                 •handle (stream) -
                 •blockStart (int) – start from here (optional)
                 •innerStart (int) – start from here (optional)
     get_block_start()
```

```
get_inner_start()
     read (size)
          read size bytes and return them
     seek (blockStart, innerStart)
class segtools.format.bgzf.writer(handle)
     Give it the handle of the stream to write to
     close()
     write (bytes)
segtools.format.fasta module
class seqtools.format.fasta.FASTA(fasta_text)
     Bases: seqtools.sequence.Seq
     FASTA()
class seqtools.format.fasta.FASTAData(data=None, file=None, dict=None)
     Slicable fast fasta It loses any additional header information in fasta header only the first non-whitespace is what
     we use
          Parameters
                 •data (bytes) - bytes of the fasta file
                 •file (string) - filename
                 •dict (dict ()) - dictionary of chromosomes
     clear()
     get_sequence (chr=None, start=None, end=None, dir=None, rng=None)
          get a sequence
              Parameters
                   •chr (string) -
                   •start (int) -
                   •end(int)-
                   •dir (char) - charcter +/-
              Parma rng
              Returnssequence
              Return typestring
     keys()
     remove (key)
class seqtools.format.fasta.FASTAFile (fname, index=None)
     Do random access with an indexed Fasta File Creates the index if its not there already
     Pre: An uncompressed fasta fileCan be called by chromosome and location slices Slices are same as array -
          zero indexed
```

1.1. Subpackages 9

Post: Makes index if doesn't exist upon being called. Can access sequence

```
Modifies: File IO reads the fasta, and writes a fasta index file
       Warning: this uses a subclass. probably should avoid that
     class Chromosome (outer, chr)
     FASTAFile.get_sequence(chr=None, start=None, end=None, dir=None, rng=None)
class seqtools.format.fasta.FASTAStream(fh, custom_buffer_size=10000000)
     Iterable Stream
          Parameters
                •fh (stream) - file handle
                •custom_buffer_size (int) - default size (10000000)
     get_entry()
     next()
seqtools.format.fastq module
class seqtools.format.fastq.FASTQ(v)
     Bases: seqtools.sequence.Seq
     fastq single entry
          Parametersv (string) – one entry
     FASTQ()
     copy()
     rc()
class seqtools.format.fastq.FASTQStream(fh)
     Iterable Stream
     get_entry()
     next()
seqtools.format.gpd module
class seqtools.format.gpd.GPD (gpd_line)
     Bases: seqtools.structure.Transcript
     This whole format is a subclass of the Transcript subclass
          Parametersgpd_line (string) -
     exons
     get_gpd_line()
          output the original gpd line Overrides Structure. Transcript
     get_line()
     get_range()
          override, we are garunteed to have the range since we initialize on reading a line
```

junctions

```
value (kev)
class seqtools.format.gpd.GPDStream(fh)
     Iterate over GPD entries
     next()
     read entry()
class seqtools.format.gpd.SortedOutputFile (filename, type='location', tempdir=None)
     a stream to write to for outputing a sorted file
          Parameters
                 •filename (string) - output file
                 •type (string) – how to sort (default location)
                 •tempdir(string)-
     close()
     write(value)
seqtools.format.psl module
Classes to work with the psl format
class seqtools.format.psl.PSL(psl_line,
                                                     reference=None,
                                                                            query_sequences=None,
                                    query_sequence=None, query_quality=None)
     Bases: seqtools.align.Alignment
     Class to define a psl line
          Parameters
                 •psl_line (string) - is a psl formated line
                 •reference – a dict/slice accessable sequences
                 •query_sequences (dict()) - a dict/slice accessable sequences
                 •query_sequence (string) - just the string that is the query sequence
     class PrivateValues
          This class was an attempt at creating closures for some values. It may be overkill for what we are doing,
          or worse, it may be slow. Values from the orignal should just be accessed though functions for consistancy
          sake. This class should remind us well that entires need to be accessed this way
          get_entry(key)
          is_entry_key(key)
          set_entries_dict(mydict)
          set_entry(key, value)
     PSL.get_PSL()
          Overrides parent to make the PSL generation just return self
     PSL.get_line()
     PSL.get_query_length()
          overrides parent to get the query length
     PSL.get_query_quality()
```

```
PSL.get_query_sequence()
Do our overrides parent to get query sequence

Return typestring

PSL.get_reference()
overrides parent to get the reference genome dict()

PSL.get_strand()
same as direction

Returnsstrand + or -
Return typechar

PSL.value(key)
Access spefific attributes of the PSL by key name. Here is how we access value by keys

Parameterskey(string) - which attribute of the PSL to get
```

### seqtools.format.sam module

Classes to work with sam and bam files

```
 \begin{array}{c} \textbf{class} \ \texttt{seqtools.format.sam.BAM} (bin\_data, & ref\_names, & fileName=None, \\ & innerStart=None, & ref\_lengths=None, \\ & line\_number=None) \\ \textbf{Bases:} \ seqtools.format.sam.SAM \end{array}
```

Very much like a sam entry but optimized for access from a bam Slows down for accessing things that need more decoding like sequence, quality, cigar string, and tags

**Warning:** Having the reference names and the reference sizes we may save some time by not reading the header each time we access the file. Theres probably a more efficent coarse to go by defining a bamfile object and having a BAMline entry being the extension of sam, and drawing most of this stuff from the bam file

#### **Parameters**

```
*bin_data (bytes) - byte data for just a single bam entry (seems unnecessary since we have the file)

*ref_names (list of names) - array of reference names

*fileName - the bam file name

*blockStart (where to begin in the file) -

*innerStart (where to begin in the decompressed block) -

*ref_lengths (dict()) - seems unnecessary to take the reference lengths because we can always get that from the header

*reference (dict()) -

*line_number (int) -

get_alignment_ranges()

return the basics for defining an alignment
get_block_start()
```

```
get_cigar()
          produce the cigar in list form
              ReturnsCigar list of [value (int), type (char)] pairs
              Return typelist
     get coord()
          get the current coordinate
              Returns[blockStart, innerStart]
              Return typelist is a pair [int, int]
     get_file_position_string()
     get_filename()
     get_inner_start()
     get_line_number()
     get_tag(key)
          retrieve the value of a single tag by its key.
            Warning: Not sure if it accommodates multiple of the same keys
     get_target_length()
          length of the entire chromosome
     value (key)
          Access basic attributes of BAM by key
class seqtools.format.sam.BAMFile (filename, blockStart=None, innerStart=None, cnt=None, refer-
                                          ence=None)
     iterable class to open and access a bam file
          Parameters
                 •filename (string) -
                 •blockStart (int) -
                 •innerStart (int) -
                 •cnt (int) -
                 •reference (dict ()) – dictionary of genomic sequences
     close()
     fetch_by_coord(coord)
          get a single entry by the coordinate location [blockStart, innerStart]
                       creates a new instance of a BAMFile object when maybe the one we had would have
           Warning:
            worked
     fetch_starting_at_coord(coord)
          starting at a certain coordinate was supposed to make output
                       creates a new instance of a BAMFile object when maybe the one we had would have
            Warning:
            worked
     get_header()
```

Warning: We already have a BGZF class, i winder why we don't put this there

#### **Parameters**

Class to define the SAM format.

### **Parameters**

### class PrivateValues

My attempt at closures again. Still think its probably not worth the trouble doing it this way. Bam files need a specific override to get\_tags and get\_cigar that would break other parts of the class if we access the variables other ways Force tags and cigars to be hidden so we don't accidently change them.

```
get_cigar()
get_entry(key)
get_tags()
is_entry_key(key)
set_cigar(cigar)
set_entries_dict(mydict)
set_entry(key, value)
set_tags(tags)
SAM.check_flag(inbit)
```

#### SAM.get SAM()

Override parent to just return itself

### SAM.get\_actual\_original\_query\_range()

This accounts for hard clipped bases and a query sequence that hasnt been reverse complemented

Returns the range covered on the original query sequence

Return typeGenomicRange

### SAM.get\_cigar()

Get list of cigar data in the form [[value1,char1],[value2,char2]...]

Returnscigar data

Return typelist of [int value, char type] pairs

#### SAM.get\_line()

assemble the line if its not there yet

Warning: this should probably not exist if the constructor takes a line

### SAM.get\_original\_query\_length()

Similar to get\_get\_query\_length, but it also includes hard clipped bases if there is no cigar, then default to trying the sequence

Returns the length of the query before any clipping

Return typeint

```
SAM.get_query_length()
```

### SAM.get\_query\_quality()

Overrides align

Warning: this returns the full query quality, not just the aligned portion

### SAM.get\_query\_sequence()

Overrides align

**Warning:** this returns the full query sequence, not just the aligned portion

### SAM.get\_range()

Necessary function for doing a locus stream For the context of a SAM file we set this to be the target range

Returnstarget range

Return typeGenomicRange

#### SAM.get\_strand()

Overrides parent to get direction from the flag

Returnsstrand/direction + or -

Return typechar

#### SAM.get tag(key)

access tags by key and get the value

**Warning:** Some of the key values may be better returned as numerical when they are. Right now i'm not sure how its implemented but probably just as a string.

```
Parameterskey – type key: string return: key rtype: string
     SAM.get_tags()
          Get all the tags
              Returnstag information
              Return typedict() of key value attributes
     SAM.get_target_length()
          Get the length of the target sequence. length of the entire chromosome :return: length :rtype: int
     SAM.get_target_range()
          Get the range on the target strand
              Returnstarget range
              Return typeGenomicRange
     SAM.is_aligned()
     SAM. value (key)
class seqtools.format.sam.SAMHeader(header_text)
     class to retain information about a SAMheader and access data from it
     get_sequence_length(sname)
     get_sequence_lengths()
     get_sequence_names()
class seqtools.format.sam.SAMStream(fh=None, minimum_intron_size=0, minimum_overhang=0,
                                            reference=None)
     minimum_intron_size greater than zero will only show sam entries with introns (junctions) minimum_overhang
     greater than zero will require some minimal edge support to consider an intron (junction)
          Parameters
                 •fh (stream) - filehandle to go through
                 •minimum_intron_size (int) - (default 0)
                 •minimum_overhang - require some minimum edge support to consider a junction. Prob-
                 ably should make more use of this
                 •reference (dict ()) – dictionary of reference sequences
     assign_handle(fh)
     get header()
          Return the object representing the header
     next()
     read_entry()
     set_junction_only (mybool=True)
class seqtools.format.sam.SamtoolsBAMStream(path,
                                                                minimum\_intron\_size=0,
                                                                                            mini-
                                                      mum_overhang=0, reference=None)
     Bases: seqtools.format.sam.SAMStream
     Stream but use samtools
     close()
```

```
seqtools.format.sam.check_flag (flag, inbit)
    Check a flag is true or false
seqtools.format.sam.is_header (line)
    true if we are in a header
seqtools.format.sam.is_junction_line (line, minlen=68, minoverhang=0)
seqtools.format.sam.sort_header (header_text)
    sort the chromosomes in a header text
```

#### **Module contents**

## seqtools.simulation package

#### **Submodules**

#### seqtools.simulation.emitter module

These classes should produce simulation products

Give it a transcriptome definition and a reference genome for itinitially give it uniform probability

#### **Parameters**

```
•transcriptome (Transcriptome) - A transcriptome from which to produce transcripts
```

```
•seed (int) – Seeded random generation
```

•rand (RandomSource) - A class that can generate randome numbers if you have one already seeded or want totally random

```
emit_transcript()
```

Get a transcript according according to weight of transcript

**Returns**One random Transcript

Return type*Transcript* 

```
set_weights_by_dict(weights)
```

input: an array of weights <<txname1> <weight1>> <<txname2> <weight2>>...if this does not get set then even weighting will be used

**Parametersweights** (list) - [[tx1,wght1],[tx2,wght2],...[txN,wightN]]

#### segtools.simulation.permute module

These classes are here to alter simulation products or other sequences

```
class seqtools.simulation.permute.MakeCuts (rand=None, seed=None)
    Class to cut the sequence to different sizes
```

#### Parameters

•rand (RandomSource) - pass a random source, otherwise it gets a new RandomSource

```
•seed (int) – if you want to set a seed here
     get_cut (seq)
     set_custom(gmin, gmu, gsigma)
          Set a minimum lengtha, and then the gaussian distribution parameters for cutting For any sequence longer
          than the minimum the guassian parameters will be used
     set_lr_cuts()
     set_sr_cuts()
class seqtools.simulation.permute.MakeErrors (rand=None, seed=None)
     Class to define how to make errors, and to introduce those errors
     random_deletion (fastq, rate)
          Perform the permutation on the sequence
              Parameters
                   •fastq(format.fastq.FASTQ) - FASTQ sequence to permute
                   •rate (float) - how frequently to permute
              ReturnsPermutted FASTQ
              Return typeformat.fastq.FASTQ
     random_flip (sequence)
          Change the direction of the sequence with 0.5 probability
     random insertion (rate, max inserts=1)
          Perform the permutation on the sequence. If authorized to do multiple bases they are done at hte rate
          defined here.
              Parameters
                   •fastq(format.fastq.FASTQ) - FASTQ sequence to permute
                   •rate (int) – how frequently to permute
                   •max_inserts – the maximum number of bases to insert (default 1)
              ReturnsPermutted FASTO
              Return typeformat.fastq.FASTQ
     random_substitution (fastq, rate)
          Perform the permutation on the sequence
              Parameters
                   •fastq(format.fastq.FASTQ) - FASTQ sequence to permute
                   •rate (float) – how frequently to permute
              ReturnsPermutted FASTQ
              Return typeformat.fastq.FASTQ
     set_after_context(base)
          Limit errors to a specific following base context
     set_before_context(base)
          Limit errors to a specific preceeding base context
     set_modified_base(base)
          Limit errors to a specific type of sequenced base
```

```
set observed base (base)
          Limit errors to a specific reference base
seqtools.simulation.permute.phred33_to_rate(q)
     Convert a phred33 character to an error rate
seqtools.simulation.permute.random_flip(sequence, rnum=None)
     Flip a sequence direction with 0.5 probability
seqtools.simulation.permute.rate_to_phred33(rate)
     Convert an error rate to a phred 33 character
seqtools.simulation.randomsource module
A class to aid in generating random numbers and sequences
class segtools.simulation.randomsource.RandomSource (seed=None)
     You can asign it a seed if you want
          Parametersseed (int) – seed the pseduorandom number generator
     choice (arr)
          Uniform random selection of a member of an list
              Parametersarr (list) – list you want to select an element from
              Returnsone element from the list
     different random nt(nt)
     gauss (mu, sigma)
          Generate a random number based on a gaussian distribution
              Parameters
                    •mu (float) – mean of distribution
                    •sigma (float) – standard deveiation of distribution (i think)
     get_weighted_random_index (weights)
          Return an index of an array based on the weights if a random number between 0 and 1 is less than an
              index return the lowest index
              Parametersweights (list) – a list of floats for how to weight each index [w1, w2, ... wN]
              Returnsindex
              Return typeint
     randint (a, b)
          Generate a random integer uniform distribution between a and b like randint of the usual random class
              Returnsrandom int between a and b
              Return typeint
     random()
          generate a random number
              Returnsuniform random float between 0 and 1
              Return typefloat
```

```
random nt()
```

Produce a random nucleotide (uniform random)

Returnsnucleotide

Return typechar

#### **Module contents**

### **Submodules**

# seqtools.align module

This module contains the most basic classes for describing and working with alignments.

```
class seqtools.align.Alignment
```

Basic class for common elements of alignments. You don't have to have a query sequence and a reference sequence to do an alignment.

```
construct_cigar (min_intron_size=68)
```

Create a CIGAR string from the alignment

ReturnsCIGAR string

Return typestring

```
get_PSL (min_intron_size=68)
```

Get a PSL object representation of the alignment.

ReturnsPSL representation

Return typePSL

```
get_SAM (min_intron_size=68)
```

Get a SAM object representation of the alignment.

**Returns**SAM representation

Return typeSAM

```
get_actual_query_range()
```

This is the actual query range for the positive strand

ReturnsRange of query positive strand covered

**Return type***GenomicRange* 

```
get_aligned_bases_count()
```

The sum of the aligned bases.

Returnslength (in base pairs)

Return typeint

```
get_alignment_ranges()
```

Return an array of alignment ranges.

```
get_alignment_strings (min_intron_size=68)
```

**Process the alignment to get information like**the alignment strings for each exon. These strings are used by the pretty print.

ReturnsString representation of the alignment in an easy to read format

**Return typestring** 

### get\_query\_length()

Warning: Must be overridden

### get\_query\_quality()

Get the quality.

Returnsquality

Return typestring

### get\_query\_sequence()

Warning: Must be overridden

### get\_reference()

Return the reference sequence

Returns reference sequence

**Return typestring** 

### get\_strand()

Warning: Must be overridden

### get\_target\_length()

Warning: Must be overridden

### get\_target\_range()

Get the range covered on the target/reference strand

ReturnsGenomic range of the target strand

**Return type***GenomicRange* 

### get\_target\_transcript (min\_intron=1)

Get the mapping of to the target strand

ReturnsTranscript mapped to target

Return typeTranscript

### print\_alignment (chunk\_size=40, min\_intron\_size=68)

print the nice looking alignment. Must have data accessable from get\_query\_sequence() and get\_reference\_sequencec()

ReturnsPretty print string.

**Return typestring** 

#### set\_query\_sequence (seq)

Assign the query sequence.

 ${f Parametersseq} \ (string) - {f sequence} \ {f of the query}$ 

### set\_reference(ref)

Set the reference sequence

**Parametersref** (string) – reference sequence

# seqtools.errors module

This module contains classes for analyzing error patterns in alignments

Its in pretty rough shape as its an early, but working, form. It works with alignqc. But it really needs some love to be a good module.

Error Analysis

I am to describe errors at several levels

Errors in the query sequence

- 1. Is a query base an error or not?
- Probability Sometimes it can be ambiguous which base is in error
- 2. What is the basic type of error?
- Mismatch
- Insertion
  - Total insertion
  - Homopolymer insertion
  - Deletion
    - \* Total deletion
      - · Before
      - · After
    - \* Homopolymer deletion
- sum of probabilities should add up to 1.)
- 3. What is the more specific error?
- · Mismatch type
- insertion/deletion Base, Length

class seqtools.errors.AlignmentErrors (alignment, min intron size=68)

Take an alignment between a target and query Uses get\_strand from alignment to orient the query All results are on the positive strand of the query (meaning may be the reverse complement of target if negative)

### **Parameters**

```
•alignment (Alignment) – alignment to be used in error calculation
•min_intron_size (int) – minmum length for an intron
```

class HPAGroup (parent, mydict)

**Homopolymer alignment group**takes a chunk of homopolymer alignment as a dictionary with 'query' and 'target' sequences set query should always be positive strand

```
Parametersmydict (dict() {'query':query sequence, 'target':target
    sequence}) - dictionary with target sequences and a parent object
```

```
get exon()
         return the exon number
     get_length()
         return the lengths of the query and the target
             Returnslengths object
             Return typedict() with {'query':query length,'target': target length}
     get_nt()
     get_quality()
         get the quality score info or false if we cannot
     get_query()
         always + strand
     get_string()
         Describe the group as a string
     get_target()
         could be + or - strand
     has_quality()
         Do we have quality score info?
     type()
AlignmentErrors.analyze quality()
     Go through HPAGroups and store the distro of ordinal values of quality scores
AlignmentErrors.close()
AlignmentErrors.get_HPAGroups()
     get a list of the HPA groups :returns: list of HPA groups :rtype: HPAGroup
AlignmentErrors.get_context_query_errors()
     A more straitfoward calculation of the context-specific errors relative to the query
         Returnsmatrix of observed contexts and values
         Return typematrix of [before][after][query]{types} with types being any base or a deletion.
AlignmentErrors.get_context_target_errors()
     A more straitfoward calculation of the context-specific errors relative to the target
         Returnsmatrix of observed contexts and values
         Return typematrix of [before][after][reference]{types} with types being any base or a deletion.
AlignmentErrors.get_general_errors()
     way to accumulate totals of error types General error report will be relative to to the total alignment length
     error rate = mismatches + insertions + deletions / alignment length
     This looks oddly written, probably should be careful not to run it twice because it looks like it would
     accumulate.
AlignmentErrors.get_quality_report_string()
     get a report on quality score distribution. currently prints to stdout
AlignmentErrors.get_query_error(i)
     Just get a single error characterization based on the index
         Parametersi (int) – list index
         Returnsbase-wise error
```

```
Return typeHPA group description
     AlignmentErrors.get_query_errors()
          Return a list of base-wise error observations for the query
               Returnslist of base-wise errors
              Return typelist of HPA groups
     AlignmentErrors.get_query_sequence()
          return the query sequence reconstructed from the descriptions
     AlignmentErrors.get_target_error(i)
          Just get a single error characterization based on the index relative to the target
               Parametersi (int) – list index
              Returnsbase-wise error
              Return typeHPA group description
     AlignmentErrors.get_target_errors()
          Just get a single error characterization based on the index relative to the target
              Parametersi (int) – list index
              Returnslist of base-wise errors
              Return typelist of HPA groups
     AlignmentErrors.get_target_sequence()
          return the target sequence reconstructed from the descriptions
     AlignmentErrors.has_quality()
          Does the current data have quality information?
class seqtools.errors.BaseError(type)
     Class for describing an error at a sinle base relative to the target or query.
     class ObservableError (type)
          Class to describe a homopolymer error or an observable insertion or deletion. Future versions of this
               should probably avoid using a nested class for this
              Parameterstype (string) – Either 'query' or target
          get_attributable_length()
              For calculating total error counts
          get changed length()
              How much the homopolymer length differs between target and query
                  Returnsabs(qlen-tlen)
                  Return typeint
          get_error_probability()
              Probability that this base is the product of an error
                   Returnsprobability
                  Return typefloat
          get_homopolymer()
              Return a class to describe the homopolymer
                  Returnshomopolymer details
                   Return typedict() return {'tseq':string,'seq':string}
```

```
get_query_base()
         Just the query base
     get_target_base()
         Just the target base
     get_type()
         get the type of the observable error
             Returnserror details
             Return typelist with 1. main type, 2. subtype, 3. details [target [nucleotide, length], query
               [nucleotide, length]]
     set (tlen, qlen, tnt, qnt)
         Set the error we are observing for the homopolymer block
             Parameters
                •tlen (int) – target homopolymer length
                •qlen (int) – query homopolymer length
                •tnt (char) – target nucleotide
                •qnt (char) – query nucleotide
class BaseError.UnobservableError(type)
     Unobservable error is a deletion for a query base an insertion for a target base A non base error has a
     probability of occuring before a base and a probability of occuring after
         Parameterstype (string) – Either 'query' or target
     get_after_probability()
     get_after_type()
     get_attributable_length()
     get_before_probability()
     get_before_type()
     get_error_probability()
     set_after (tlen, qlen, nt, p)
     set_before (tlen, qlen, nt, p)
BaseError.get_adjusted_error_count()
     Get the total error count associated with this single base. This would typically be one but sometimes it
         may be larger for instertions.
         Returnserror count
         Return typefloat
BaseError.get_base()
     Get the single base at this position.
         Returnsbase
         Return typechar
BaseError.get_error_probability()
     This means for the base we are talking about how many errors between 0 and 1 do we attribute to it?
         For the 'unobserved' errors, these can only count when one is adjacent to base
```

**Returns**error probability p(error observed)+(1-p error observed)\*error unobserved

### Return typefloat

```
BaseError.get_homopolymer()
```

Return the hompolymer on target and the homopolymer on query assicated with this base

**Returns**homopolymer dict {tseq:sequence,qseq:sequence}

Return typedict()

#### BaseError.get observable()

Get error information that can be seen

ReturnsObservable error object

Return typeObservableError

#### BaseError.get\_observable\_error\_probability()

get the probability of an observable error occuring at a base

Returnserror probability

Return typefloat

```
BaseError.get_string()
```

Get a string representation of this single base error.

Returnsreport

Return typestring

```
BaseError.get_unobservable()
```

Unobservable errors inferred, like if its relative to the target and an insertion, then it is not observed in the target, we just know it was inserted between two bases in the target.

ReturnsUnobservable error object

Return type*UnobservableError* 

### BaseError.get\_unobservable\_error\_probability()

get the probability of an unobservable error occuring at a base

Returnserror probability

Return typefloat

```
BaseError.is_any_error()
```

If theres any reason to attribute this base to an error return True otherwise false

Returnsthere\_is\_error

Return typebool

```
BaseError.set_observable(tseq, qseq)
```

Set the observable sequence data

#### **Parameters**

```
•tseq(string) - target sequence (from the homopolymer)
```

•qseq (string) – query sequence (from the homopolymer)

```
BaseError.set_unobserved_after(tlen, qlen, nt, p)
```

Set the unobservable sequence data after this base

#### **Parameters**

•tlen (int) - target homopolymer length

```
•qlen (int) – query homopolymer length
```

•nt (char) - nucleotide

•p (float) – p is the probability of attributing this base to the unobserved error

### BaseError.set\_unobserved\_before (tlen, qlen, nt, p)

Set the unobservable sequence data before this base

#### **Parameters**

```
•tlen (int) – target homopolymer length
```

•qlen (int) – query homopolymer length

•nt (char) - nucleotide

•p (float) - p is the probability of attributing this base to the unobserved error

### class seqtools.errors.ErrorProfileFactory

This class is used to create an error profile. It doesn't require any special input to create a new instance of it. You add to it with the add\_alignment() function.

#### add\_alignment(align)

Calculate alignment errors from the alignment and add it to the profile.

### add\_alignment\_errors (ae)

If you alread have thealignment errors, add them for profile construction.

#### close()

Set some objects to None to hopefully free up some memory.

### combine\_context\_errors()

Each alignment contributes some information to the error report. These reports for each alignment need to be gone through and combined into one report.

**Returns**Dictionary containing the error counts on context base

Return typedict()

### get\_alignment\_errors()

Return an object that describes the errors

Returns Alignment Errors

Return typeGeneralErrorStats

### get\_min\_context\_count (context\_type)

Calculate out which context has the minum coverage thusfar.

```
Parameterscontext_type (string) - 'target' or 'query'
```

**Returns**Minimum Coverage

Return typeint

### get\_query\_context\_error\_report()

Get a report on context-specific errors relative to what is expected on the query strand.

**Returns**Object with a 'header' and a 'data' where data describes context: before,after ,reference, query. A total is kept for each reference base, and individual errors are finally checked

Return typedict()

### get\_query\_context\_errors()

Return the query context errors

```
ReturnsDictionary containing the error counts on context base
```

```
Return typedict()
```

```
get_string()
```

Make a string reprentation of the error stats.

Returnserror profile

Return typestring

```
get_target_context_error_report()
```

Get a report on context-specific errors relative to what is expected on the target strand.

**Returns**Object with a 'header' and a 'data' where data describes context: before,after ,reference, query. A total is kept for each reference base, and individual errors are finally checked

**Return type**dict()

```
get_target_context_errors()
```

Return the target context errors

**Returns**Dictionary containing the error counts on context base

**Return type**dict()

```
write_context_error_report (file, context_type)
```

Write a context error report relative to the target or query into the specified filename

#### **Parameters**

```
•file (string) – The name of a file to write the report to
```

•context\_type (string) - They type of profile, target or query based

### class seqtools.errors.GeneralErrorStats

Keep track of general errors across the length of an alignment

### add\_alignment\_errors (ae)

Add alignment errors to the group

#### **Parameters**

```
•ae – one set of alignment errors
```

```
•type -
```

```
get_report()
```

Another report, but not context based

```
get_stats()
```

Return a string describing the stats

```
get_string()
```

make a string representation of the general error report

# seqtools.graph module

This module has classes to provide graph structures and graph-based operations.

```
class seqtools.graph.Edge (node1, node2, directionless=False, weight=None)
    Class defines an edge.
```

directed graph by default

```
Parameters
                 •node1 (Node) – required - node 2
                  •node2 (Node) – required - node 1
                  •directionless (bool) – by defalt we are directed graph
                 •weight - value to weight the edge
     get id()
          get the internal id of the edge. probably uuid4
     get_node1()
          get what is called node1
              Returnsnode1
              Return typeNode
     get_node2()
          get what is called node2
              Returnsnode2
              Return typeNode
     get_node_ids()
          get the uuid4 ids of the nodes in the edge
              Returnslist of [id1,id2]
              Return typelist
     get_weight()
          get the weight if its been set
     is_directionless()
          get the direction status of the edge
     set_weight (weight)
          can set weight to some number
class segtools.graph.Graph (directionless=False)
     Graph basic structure.
     Use directed graph by default
          Parametersdirectionless (bool) – use an undirected graph if set to true
     add edge (edge, verbose=True)
          add an edge to the graph
              Parameters
                    •edge (Edge) -
                    •verbose (bool) -
                      -optional default (True)
     add_node (node)
          add a node to the graph
              Parametersnode (Node) -
     connected_nodes (node, exclude_ids=None)
          get all the connected nodes
```

```
Parameters
              •node (Node) -
              •exclude_ids(list or None)-
         Returns list of connected nodes
         Return typeNode[]
find_cycle()
     return a single cycle, greedy first one found in terms of nodes return as an array of nodes or None. done
         by depth first search through nodes
         Returnsnodes in the cycle (list) or None
         Return typeNodes[] or None
get_children (node)
     Find all the children of a node. must be a undirectional graph with no cycles
         Parametersnode (Node) -
         Returnslist of nodes
         Return typeNode[]
get directed paths from node(node, prev=[])
     get all the paths in terms of lists of nodes from a node. needs to be a directed graph with no cycles.
         Parameters
              •node (Node) -
              •prev (list) – do not used, used by the class when calling it recurrsively
get_edges()
     a list of edges
         Returnsedges
         Return typeEdge[] list
get_node_edges (node, type='both')
     given a node return the edges attached, by default get both incoming and outgoing
         Parameters
              •node (Node) -
              •type(string - default 'both')-
         Returnsedge list
         Return typeEdge[] edge list
get_nodes()
     a list of the nodes
         ReturnsNodes
         Return typeNode[] list of nodes
get_report()
     describe the graph
         Returnsreport
```

```
Return typestring
     get_roots()
          get the roots of a graph. must be a directed graph
              Returnsroot list of nodes
              Return typeNode[]
     get_status_string()
          get a string describing some stats about a graph
     merge_cycles()
          remove cycles by mergine cyclic nodes into single nodes their payloads are added to a list
     partition_graph (verbose=False)
          break a graph into multiple graphs if they are not connected
              Returnslist of graphs
              Return typeGraph[]
     remove edge (edge)
          remove an edge from the graph
              Parametersedge (Edge) -
     remove\_node(node)
          remove a node from the graph
              Parametersnode (Node) -
class seqtools.graph.Node(payload=None)
     Class to describe a node
          Parameterspayload (anything you want) – Empty payload by default
     get_id()
          return the uuid4 id
     get_payload()
          return whats curently held in payload
     set payload(payload)
          set the payload to anything you want
```

# seqtools.range module

These classes are to help deal with genomic coordinates and things associated with those coordinates.

```
class seqtools.range.Bed (chrom, start, finish, dir=None)
    Bases: seqtools.range.GenomicRange
```

**Bed format is a chromosome, start (0-index), end (1-index).** It is a child of GenomicRange but modifies the class to use the 0-based start 1-based end style of a bed file

### **Parameters**

```
chrom (string) -start (int) - 0-indexedfinish (int) - 1-indexed
```

```
•dir(char)-
                    -or - (optional)
     copy()
          Override copy to make another copy
               Returnsa new copy of this object
               Return typeBed
class seqtools.range.BedArrayStream(bedarray)
     Make a stream from a bedarray
     Read as an interator or with read_entry()
          Parametersbedarray (Bed[]) -
     next()
          call read_entry() from inside this iterator
     read_entry()
          get the next value from the array, and set internal iterator so next call will be next entry
               ReturnsThe next GenomicRange entry
               Return typeGenomicRange
class segtools.range.BedStream(fh)
     Make a stream from a handle, keep it as an iterator
          Parametersfh (handle) – readable file handle or stream
     next()
     read_entry()
          read the next bed entry from the stream
class seqtools.range.GenomicRange(chr=None,
                                                           start=None,
                                                                            end=None,
                                                                                            dir=None,
                                            range_string=None)
     A basic class for keeping genomic range data. \overline{\text{It}} is 1-indexed for both start and end.
          Parameters
                  •chr (char) - chromosome name
                  •start (int) – 1-indexed starting base
                  •end (int) – 1-indexed ending base
                  •dir (char) - direction
                  •range_string (string) – set from string like chr5:801-900
     adjacent (rng2, use direction=False)
          Returns true if the two beds are directly next to eachother, 'touching'
               Parameters
                    •rng2 -
                    •use_direction - false by default
                    •type - GenomicRange
                    •type - use_direction
```

```
cmp (range2, overlap_size=0)
     the comparitor for ranges
        •return 1 if greater than range2
        •return -1 if less than range2
        •return 0 if overlapped
         Parameters
              •range2 (GenomicRange) -
              •overlap_size (int) – allow some padding for an 'equal' comparison (default 0)
copy()
     Create a new copy of selfe. does not do a deep copy for payload
         Returnscopied range
         Return typeGenomicRange
distance (rng)
     The distance between two ranges.
         Parametersrng (GenomicRange) – another range
         Returnsbases separting, 0 if overlapped or adjacent, -1 if on different chromsomes
         Return typeint
dump_serialized()
     dump the pickle for this object
         Returnspickled object
         Return typepickle_object
equals (rng)
get_bed_array()
     Return a basic three meber bed array representation of this range
         Returnslist of [chr,start (0-indexed), end (1-indexed]
         Return typelist
get_bed_coordinates()
     Same as get bed array. These are the 0-indexed start, 1-indexted stop coordinates
         Returnsbed array [chr,start-1, end]
get_direction()
     return the direction
         Returnsthe direction or strand +/- (or None if not set)
         Return typechar
get_genomic_coordinates()
     These are the 1-indexed coordinates in list form
         Returnslist of coords [chr, start (1-indexed), end(1-indexed)
         Return typelist
```

```
get_payload()
    Returns the payload, whatever it may be
get_range()
     For compatability with some range-based tools that need to call this function
         Returnsthis object
         Return typeGenomicRange
get_range_string()
     get the range string representation. similar to the default input for UCSC genome browser
         Returns representation by string like chr2:801-900
         Return typestring
length()
     get the length of the range
         Returnslength
         Return typeint
load serialized(instr)
    load a pickeled range back into the object
         Parametersinstr (pickled_object) -
merge (range2, use direction=False)
     merge this bed with another bed to make a longer bed. Returns None if on different chromosomes or
     direction is set true and they are in differet strands.
         Parameters
              •range2 (GenomicRange) -
              •use_direction (bool) – consider direction for overlapping? (default False)
         Returnsbigger range with both
         Return typeGenomicRange
overlap_size(in_genomic_range)
     The size of the overlap
         Parametersin_genomic_range (GenomicRange) - the range to intersect
         Returns count of overlapping bases
         Return typeint
overlaps (in_genomic_range, use_direction=False, padding=0)
     do the ranges overlap?
         Parameters
              •in_genomic_range (GenomicRange) - range to compare to
              •use_direction (bool) - default (False)
              •padding (int) – add to the ends this many (default 0)
         ReturnsTrue if they overlap
         Return typebool
```

```
Does the range overlap with a padded range. Looks like this is fairly redundant with the overlaps function
               Parameters
                    •in_genomic_range (GenomicRange) - range to compare to
                    •padding (int) – amount to add onto ends to search
               Returnstrue if they overlap, false if they do not
               Return typebool
     print_range()
          print the range string to stdout
     set_direction(dir)
          set he direction
               Parametersdir (char) - direction + or -
     set_payload(inpay)
          Set the payload. Stored in a list to try to keep it as a reference
               Parametersinpay – payload input - any type that can be pushed into a list
     subtract (range2, use_direction=False)
          Take another range, and list of ranges after removing range2, no garuntees on payload
               Parameters
                    •range2 (GenomicRange) -
                    •use_direction (bool) -
               ReturnsList of Genomic Ranges
               Return typeGenomicRange[]
     union (range2)
          Intersection may be a better description. Return the chunk they overlap as a range. direction is destroyed
               Parametersrange2 (GenomicRange) -
               ReturnsRange with the intersecting segement, or None if not overlapping
               Return typeGenomicRange
class seqtools.range.Loci
     multiple locus combined together when new members are added based on parameters
     add locus (inlocus)
          Adds a locus to our loci, but does not go through an update our locus sets yet
     merge_down_loci()
          Called internally to make loci overlapping into one set
     set_minimum_distance(over)
          In preparation for combining loci specify how many basepairs they may be separated by but still get merged
     set_use_direction(inbool)
          Do we want to only combine loci when they have the same direction, if so, set to True
     update_loci()
          Goes through and combines loci until we have one set meeting our overlap definition
```

overlaps\_with\_padding(in\_genomic\_range, padding)

```
class segtools.range.Locus
     A Locus is a colloction of GenomicRanges that fall within some distance of one another
     add_member (grange)
          Add a genomic range to the locus
              Parametersgrange (GenomicRange) -
     set_use_direction(inbool)
          Set to true if you want all locus members to share the same direction
              Parametersinbool (bool) -
seqtools.range.merge_ranges (inranges, already_sorted=False)
     from a list of genomic range or bed entries, whether or not they are already sorted, make a flattend range
          list of ranges where if they overlapped, they are now joined (not yet) The new range payloads will be the
          previous ranges
          Parameters
                 •inranges (GenomicRange[]) -
                  •already_sorted (bool) – has this already been sorted (defaults to False)
          Returns sorted ranges
          Return typeGenomicRange[]
seqtools.range.pad_ranges(inranges, padding, chr_ranges=None)
     Add the specfied amount onto the edges the transcripts
          Parameters
                  •inranges (GenomicRange []) - List of genomic ranges in Bed o GenomicRange for-
                  mat.
                  •padding (int) - how much to add on
                  •chr_ranges – looks like the list of ranges within which to pad
seqtools.range.ranges_to_coverage(rngs, threads=1)
     take a list of ranges as an input output a list of ranges and the coverage at each range :param rngs: bed ranges
     on a single chromosome. not certain about that single chromosome requirement :type rngs: GenomicRange[] or
     Bed[]:param threads: Not currently being used:type threads: int
          Returnsout is the non-overlapping bed ranges with the edition of depth
          Return typeGenomicRange[]
seqtools.range.sort_genomic_ranges(rngs)
     sort multiple ranges
seqtools.range.sort_ranges(inranges)
     from an array of ranges, make a sorted array of ranges
```

Parametersinranges (GenomicRange []) - List of GenomicRange data

**Returns**a new sorted GenomicRange list

**Return type**GenomicRange[]

```
seqtools.range.string_to_genomic_range(rstring)
```

Convert a string to a genomic range

**Parametersrstring** – string representing a genomic range chr1:801-900

### **Returns**object representing the string

```
Return typeGenomicRange
```

```
seqtools.range.subtract_range_array (bed1, beds2, is_sorted=False)
subtract several ranges from a range, returns array1 - (all of array2)
```

### **Parameters**

```
    bed1 (Bed or GenomicRange) -
    beds2 (Bed[] or GenomicRange[]) - subtract all these beds from bed1
    is_sorted - has it been sorted already? Default (False)
    is sorted - bool
```

seqtools.range.subtract\_ranges (rls, r2s, already\_sorted=False)
Subtract multiple ranges from a list of ranges

### **Parameters**

```
    r1s (GenomicRange[]) - range list 1
    r2s (GenomicRange[]) - range list 2
    already_sorted - default (False)
```

**Returns**new range r1s minus r2s

Return typeGenomicRange[]

seqtools.range.union\_range\_array (bed1, beds2, payload=None, is\_sorted=False)

Does not do a merge if the payload has been set

### **Parameters**

```
bed1 (GenomicRange) -bed2 (GenomicRange) -
```

•payload (int) – payload=1 return the payload of bed1 on each of the union set, payload=2 return the payload of bed2 on each of the union set, payload=3 return the payload of bed1 and bed2 on each of the union set

```
•is_sorted(bool)-
```

## seqtools.sequence module

Returnscopy of the sequence

Return typeSeq

```
fasta()
          Get the fasta formated string Pre: seq and name are set Post: string representation of the fasta entry
              Returnsfasta string
              Return typestring
     gc content()
          Calculate the GC content of the sequence
              ReturnsGC content
              Return typefloat
     n_count()
          Count the numbers of 'N's in a sequence. case insensitive.
              ReturnsN count
              Return typeint
     rc()
          reverse complement
              Returnsreverse complemented sequence of same name
              Return typeSeq
seqtools.sequence.decode_name(safename)
     Make an encoded name into its decoded.
          Parameterssafename – thing to be decoded
          Returnsdecoded name
          Return typestring
seqtools.sequence.encode_name(conversion_string)
     Make a name into an encoding that can store any character
          Parametersconversion_string (string) – thing to be encoded
          Returnsencoded_name
          Return typestring
seqtools.sequence.rc(seq)
     Fast reverse complement function using a translation table and slice
          Parametersseq (string) – string to reverse complement
          Returnsreverse complemented sequence
          Return typestring
seqtools.statistics module
This module contains many list-based functions to calculate descriptive statistics.
```

```
seqtools.statistics.N50 (arr)
N50 often used in assessing denovo assembly.

Parametersarr (number[] a number array) - list of numbers
ReturnsN50
```

```
Return typefloat
seqtools.statistics.average(arr)
     average of the values, must have more than 0 entries.
          Parametersarr (number[] a number array) - list of numbers
          Returnsaverage
          Return typefloat
seqtools.statistics.median(arr)
     median of the values, must have more than 0 entries.
          Parametersarr (number[] a number array) – list of numbers
          Returnsmedian
          Return typefloat
seqtools.statistics.standard_deviation(arr)
     standard deviation of the values, must have 2 or more entries.
          Parametersarr (number[] a number array) - list of numbers
          Returnsstandard deviation
          Return typefloat
seqtools.statistics.variance(arr)
     variance of the values, must have 2 or more entries.
          Parametersarr (number[] a number array) - list of numbers
          Returnsvariance
          Return typefloat
segtools.stream module
Classes to help stream biological data
class seqtools.stream.GZippedOutputFile (filename)
     use gzip utility to compress output
          Parametersfilename (string) – filename to write to
     close()
     write(value)
class seqtools.stream.LocusStream(stream)
     Works for any stream with ordered range bound objects that have the functions. Locus Stream is a stream
          itself, and is iterable
        1.read_entry()
        2.get_range()
     Data is not stored as an actual Locus object, but rather in list in the payload of the range covered by the locus
```

next()

**Parametersstream** (Stream) – ordered stream with range

```
read_entry()

As long as entires overlap keep putting them together in a list that is the payload for a range that describes the bounds of the list

Returns range with payload list of elements

Return type Genomic Range

class seqtools.stream.MultiLocusStream (streams)

Take an array streams Each element should be sorted by position Streams need to have this method:

1.read_entry()

2.get_range()

Parametersstreams (list) - list of streams

next()

read_entry()

get the next aggrogate of streams

Returns range containing a list of entries from each stream that are from the overlapping part Return type Genomic Range
```

## seqtools.structure module

```
this collection of classes helps us operate on mappings of transcripts
class seqtools.structure.Exon (rng=None)
     class to describe an exon
          Parametersrng (GenomicRange) -
     dump serialized()
     get_length()
     get_range()
     load_serialized(instr)
     set is leftmost(boo=True)
     set_is_rightmost(boo=True)
     set_left_junc(junc)
     set_right_junc(junc)
class seqtools.structure.Junction(rng_left=None, rng_right=None)
     class to describe a junction
          Parameters
                •rng_left (GenomicRange) - left side of junction
                •rng_right (GenomicRange) - right side of junction
     cmp (junc, tolerance=0)
          output comparison and allow for tolerance if desired
```

```
•-1 if junc comes before self
        •1 if junc comes after self
        •0 if overlaps
        •2 if else
         Parameters
              •junc (Junction) -
              •tolerance (int) – optional search space (default=0, no tolerance)
         Returnsvalue of comparison
         Return typeint
dump_serialized()
     Get string representation of the junction
         Returnsserialized object
         Return typestring
equals (junc)
     test equality with another junction
get left exon()
     get the exon to the left of the junction
         Returnsleft exon
         Return typeExon
get_range_string()
     Another string representation of the junction. these may be redundant.
get_right_exon()
     get the exon to the right of the junction
         Returnsright exon
         Return typeExon or GenomicRange
get_string()
     A string representation of the junction
         Returns string representation
         Return typestring
load serialized(instr)
     load the string
         Parametersinstr (string) -
```

# set\_exon\_right (ex) assign the right exon

set\_exon\_left (ex)
assign the left exon

overlaps (junc, tolerance=0)

see if junction overlaps with tolerance

```
set left(rng)
           Assign the leftmost range
     set_right (rng)
           Assign the right most range
class segtools.structure.Transcript
     Class to describe the mapping of a basic transcript
     class ExonOverlap (self1, tx obj1, tx obj2, multi minover=10, multi endfrac=0, multi midfrac=0.8,
                           single_minover=50, single_frac=0.5, multi_consec=True)
           class to describe exon overlap
               Parameters
                    •tx-
                     •multi_minover (int) - multi-exons need to overlap by at lest this much to be consid-
                     ered overlapped (default 10)
                     •multi_endfrac (float) - multi-exons need an end fraction coverage of at least this
                     by default (default 0)
                     •multi_midfrac(float) - multi-exons need (default 0.8) mutual coverage for internal
                     exons
                    •single_frac (float) – at least this fraction of single exons must overlap (default 0.5)
               Parma single_minoversingle-exons need at least this much shared overlap (default 50)
               Parma multi consecexons need to have multiexon consecutive mapping to consider it a match
                   (default True)
               ReturnsExonOverlap report
               Return typeTranscript.ExonOverlap
           analyze_overs (self1)
               A helper function that prepares overlap and consecutive matches data
           calculate_overlap(self1)
               Create the array that describes how junctions overlap
           consecutive_exon_count (self1)
               Best number of consecutive exons that overlap
                   Returnsmatched consecutive exon count
                   Return typeint
           is compatible (self1)
               Return True if the transcripts can be combined together
                   Returnscan be combined together
                   Return typebool
           is_full_overlap(self1)
               true if they are a full overlap
                   Returnsis full overlap
                   Return typebool
           is_subset (self1)
               Return value if tx_obj2 is a complete subset of tx_obj1 or tx_obj1 is a complete subset of tx_obj2
                    •Return 1: Full overlap (mutual subests)
                     •Return 2: two is a subset of one
```

```
•Return False if neither is a subset of the other
             Returns subset value
             Return typeint
     match_exon_count (self1)
         Total number of exons that overlap
             Returnsmatched exon count
             Return typeint
class Transcript . JunctionOverlap (self1, tx_obj1, tx_obj2, tolerance=0)
     Class for describing the overlap of junctions between transcripts
     This should probably be not a child.
         Parameters
               •tx_obj1 (Transcript) - transcript1
               •tx_obj2 (Transcript) - transcript2
               •tolerance (int) – how far before its no longer a matched junction
     analyze_overs (self1)
         A helper function to prepare values describing overlaps
     calculate_overlap(self1)
         Create the array that describes how junctions overlap
     is compatible (self1)
         Return True if the transcripts can be combined together
             ReturnsTrue if we can combine
             Return typebool
     is_full_overlap(self1)
         True if its a full overlap
             ReturnsTrue if its a full overlap
             Return typebool
     is_subset (self1)
         Return value if tx obj2 is a complete subset of tx obj1 or tx obj1 is a complete subset of tx obj2
         values:
            •Return 1: Full overlap (mutual subests)
            •Return 2: two is a subset of one
            •Return 3: one is a subset of two
            •Return False if neither is a subset of the other
     match_junction_count (self1)
Transcript.copy()
     A copy of the transcript
         Returnstranscript copy
         Return typeTranscript
Transcript.dump_serialized()
     Generate a string representation of the transcript
         Returnsserialized object
         Return typestring
```

•Return 3: one is a subset of two

```
Transcript.exon_overlap(tx, multi_minover=10, multi_endfrac=0, multi_midfrac=0.8, sin-
                                gle minover=50, single frac=0.5, multi consec=True)
     Get a report on how mucht the exons overlap
         Parameters
              •tx-
              •multi_minover (int) - multi-exons need to overlap by at lest this much to be consid-
               ered overlapped (default 10)
              •multi endfrac (float) – multi-exons need an end fraction coverage of at least this
               by default (default 0)
              •multi_midfrac(float) - multi-exons need (default 0.8) mutual coverage for internal
              •single_frac (float) – at least this fraction of single exons must overlap (default 0.5)
         Parma single_minoversingle-exons need at least this much shared overlap (default 50)
         Parma multi_consecexons need to have multiexon consecutive mapping to consider it a match
             (default True)
         ReturnsExonOverlap report
         Return typeTranscript.ExonOverlap
Transcript.exons
     Maybe the most core property of the transcript are the exon defintions. This is an array of exons.
Transcript.get_chrom()
     the reference chromosome. greedy return the first chromosome in exon array
         Returnschromosome
         Return typestring
Transcript.get_exon_count()
     Count the exons in the transcript
         Returnsexon count
         Return typeint
Transcript.get_fake_gpd_line()
     Convert a mapping to a fake GPD line. not sure why its called fake
         Returnsgpd line
         Return typestring
Transcript.get_fake_psl_line(ref)
     Convert a mapping to a fake PSL line
         Parametersref (dict()) – reference genome dictionary
         Returnspsl line
         Return typestring
Transcript.get_gene_name()
     retrieve the gene name
         Returnsgene name
```

Return typestring

```
Transcript_qet_qpd_line(transcript_name=None, gene_name=None, strand=None)
     Get the genpred format string representation of the mapping
         Parameters
              •transcript_name (string) -
              •gene name (string) -
              •strand(string)-
         ReturnsGPD line
         Return typestring
Transcript.get_id()
     Return a unique id created for this transcript when it was made
         Returnsuuid4 id as a string
         Return typestring
Transcript.get_junction_string()
     Make a string representation of all the junctions
         Returnsjunctions as a string
         Return typestring
Transcript.get junctions string()
     Get a string representation of the junctions. This is almost identical to a previous function.
     That function is get_junction_string. A refactor should clear this redundancy.
         Returns string representation of junction
         Return typestring
Transcript.get_length()
     Return the length of the transcript in bp. Its the sum of the exons
         Returnslength
         Return typeint
Transcript.get_payload()
     Get the payload currently being stored
         Returnspayload
         Return typeanything that can be stored in a list
Transcript.get_range()
     Get the range from the leftmost exon to the rightmost
         Returnstotal range
         Return typeGenomicRange
Transcript.get_sequence(ref_dict=None)
     A streutre is defined so get, if the sequence is not already there, get the sequence from the reference
         Parametersref_dict (dict ()) - reference dictionary (only necessary if sequence has not
            been set already)
Transcript.get strand()
```

Get the strand

```
Returnsdirection + or -
         Return typechar
Transcript.get_transcript_name()
     retrieve the transcript name
         Returnstranscript name
         Return typestring
Transcript.junction_overlap(tx, tolerance=0)
     Calculate the junction overlap between two transcripts
         Parameters
              •tx (Transcript) - Other transcript
              •tolerance (int) – how close to consider two junctions as overlapped (default=0)
         ReturnsJunction Overlap Report
         Return typeTranscript.JunctionOverlap
Transcript.junctions
     Can be inferred from the exons, this is an array of junctions
Transcript.load_serialized(instr)
     Load a serialized object string into the object
         Parametersinstr (string) – The serialized string
Transcript.overlap_size(tx2)
     Return the number of overlapping base pairs between two transcripts
         Parameterstx2 (Transcript) - Another transcript
         Returnsoverlap size in base pairs
         Return typeint
Transcript.set_exons_and_junctions_from_ranges(rmgs)
     set all exons and subsequestly juntions from these exon ranges; does not set direction of transcript;
         ranges need to be ordered in target order left to right
     This is a core feature for setting up a transcript.
         Parametersrngs (GenomicRange[]) - A list of ranges ordered left to right
Transcript.set_gene_name (name)
     assign a gene name
         Parametersname (string) – name
Transcript.set_payload(val)
     Set a payload for this object
         Parametersval (Anything that can be put in a list) - payload to be stored
Transcript.set_range()
     Use the exons that are already present to set the range. In a refactor this seems like it should go away or
     become private.
Transcript.set_sequence (ref_dict)
     use the reference dictionary to set the transcript's sequence
         Parametersref dict (dict ()) – reference dictionary
```

```
Transcript.set_strand(dir)
          Set the strand (direction)
              Parametersdir (char) - direction + or -
     Transcript.set_transcript_name (name)
          assign a transcript name
              Parametersname (string) - name
     Transcript.smooth_gaps (min_intron)
          any gaps smaller than min_intron are joined, andreturns a new transcript with gaps smoothed
              Parametersmin_intron (int) – the smallest an intron can be, smaller gaps will be sealed
              Returns a mapping with small gaps closed
              Return typeTranscript
     Transcript.subset (start, finish)
          Make a trimmed transcriptPre: Start base index 0 Post: Finish base index 1
              Parameters
                    •start (int) - 0-index start
                    •end(int)-
              Returnssubset transcript
              Return type Transcript
     Transcript.union(tx2)
          Find the union, or perhaps intersection is a better word for it, for two transcripts. This makes a new
          transcript.
              Parameterstx2 (Transcript) - transcript 2
              Returns overlapping portion of the transcripts
              Return typeTranscript
     Transcript.validate()
          be certain the scructure is a transcriptome
              Returnstrue if exon order is compatible with a transcriptome
              Return typelist
class segtools.structure.TranscriptGroup
     A transcript group is like the fuzzy gpd class we had before
     class JunctionGroup (self1, outer)
          Describe a junction as a group of junctions with options for junction tolerance
          add_junction (self1, tx_index, junc_index, tolerance=0)
              add a junction
          get_junction(self1)
              return the consensus junction
     TranscriptGroup.add_transcript(tx, juntol=0, verbose=True)
     TranscriptGroup.get_transcript(exon_bounds='max')
          Return a representative transcript object
```

```
class seqtools.structure.TranscriptLoci
     combine together compatible multiple transcript groups to form a simpler set of transcripts
     add_transcript (tx)
          Add a transcript to the locus
             Parameterstx (Transcript) - transcript to add
     get_depth_per_transcript (mindepth=1)
          using all the transcripts find the depth
     get_range()
          Return the range the transcript loci covers
              Returnsrange
             Return typeGenomicRange
     get_transcripts()
          a list of the transcripts in the locus
     partition loci(verbose=False)
          break the locus up into unconnected loci
              Returnslist of loci
             Return typeTranscriptLoci[]
     remove transcript(tx id)
          Remove a transcript from the locus by its id
             Parameterstx_id(string) -
     set_merge_rules(mr)
          Define rules for how to merge transcripts
             Parametersmr (TranscriptLociMergeRules) -
class seqtools.structure.TranscriptLociMergeRules (merge_type)
     Establish rules up on which to merge loci
     get_exon_rules()
     get_juntol()
     get_merge_type()
     get_use_junctions()
     get_use_multi_exons()
     get_use_single_exons()
     set_juntol(juntol)
     set_use_junctions(boo=True)
class seqtools.structure.Transcriptome (gpd_file=None, ref_fasta=None)
     a class to store a transcriptome
          Parameters
                •gpd_file (string) – filename
                •ref_fasta(dict())-
     add_transcript (transcript)
```

```
dump_serialized()
get_transcripts()
load_serialized(instr)
```

## **Module contents**

1.11. Module contents 49

# CHAPTER 2

# Indices and tables

- genindex
- modindex
- search

## S

```
seqtools, 49
seqtools.align, 20
seqtools.cli,6
seqtools.cli.cli_front,5
seqtools.cli.utilities,5
seqtools.cli.utilities.bam_bgzf_index,
seqtools.cli.utilities.bam_to_bed_depth,
seqtools.cli.utilities.fasta_to_fake_fastq,
seqtools.cli.utilities.fasta_to_tsv,4
seqtools.cli.utilities.fastq_to_fasta,
      4
seqtools.cli.utilities.fastq_to_tsv,4
seqtools.cli.utilities.trim_fasta,5
seqtools.cli.utilities.trim_fastq,5
seqtools.cli.utilities.tsv_to_fasta,5
seqtools.cli.utilities.tsv_to_fastq,5
seqtools.errors, 22
seqtools.format, 17
seqtools.format.bamindex, 6
seqtools.format.bed, 8
seqtools.format.bgzf, 8
seqtools.format.fasta, 9
seqtools.format.fastq, 10
seqtools.format.gpd, 10
seqtools.format.psl, 11
seqtools.format.sam, 12
seqtools.graph, 28
seqtools.range, 31
seqtools.sequence, 37
seqtools.simulation, 20
seqtools.simulation.emitter, 17
seqtools.simulation.permute, 17
seqtools.simulation.randomsource, 19
seqtools.statistics, 38
seqtools.stream, 39
seqtools.structure, 40
```

54 Python Module Index

A	BaseError (class in seqtools.errors), 24
add_alignment() (seqtools.errors.ErrorProfileFactory method), 27	BaseError.ObservableError (class in seqtools.errors), 24 BaseError.UnobservableError (class in seqtools.errors),
add_alignment_errors() (seq-	25
tools.errors.ErrorProfileFactory method),	Bed (class in seqtools.range), 31
27	Bed12 (class in seqtools.format.bed), 8
add_alignment_errors() (seq-	BedArrayStream (class in seqtools.range), 32
tools.errors.GeneralErrorStats method), 28	BedStream (class in seqtools.range), 32
add_edge() (seqtools.graph.Graph method), 29	BGZF (class in seqtools.format.sam), 14
add_junction() (seqtools.structure.TranscriptGroup.Junctio	n <b>G</b> roup
method), 47	<b>C</b> .
add_locus() (seqtools.range.Loci method), 35	calculate_overlap() (seq-
add_member() (seqtools.range.Locus method), 36	tools.structure.Transcript.ExonOverlap
add_node() (seqtools.graph.Graph method), 29	method), 42
add_transcript() (seqtools.structure.TranscriptGroup	calculate_overlap() (seq-
method), 47	tools.structure.Transcript.JunctionOverlap
add_transcript() (seqtools.structure.TranscriptLoci	method), 43
method), 48	check_flag() (in module seqtools.format.bamindex), 7
add_transcript() (seqtools.structure.Transcriptome	check_flag() (in module seqtools.format.sam), 16
method), 48	check_flag() (seqtools.format.sam.SAM method), 14
adjacent() (seqtools.range.GenomicRange method), 32	check_ordered() (seqtools.format.bamindex.BAMIndex
Alignment (class in seqtools.align), 20	method), 6
AlignmentErrors (class in seqtools.errors), 22	choice ()  (seqtools. simulation. random source. Random Source  (seqtools. simulation. random source)  (seqtools. simulat
AlignmentErrors.HPAGroup (class in segtools.errors), 22	method), 19
analyze_overs() (seqtools.structure.Transcript.ExonOverlap	clear() (seqtools.format.fasta.FASTAData method), 9
method), 42	close() (seqtools.errors.AlignmentErrors method), 23
analyze_overs() (seqtools.structure.Transcript.JunctionOve	rialose() (seqtools.errors.ErrorProfileFactory method), 27
method), 43	close() (seqtools.format.bgzf.writer method), 9
analyze_quality() (seqtools.errors.AlignmentErrors method), 23	close() (seqtools.format.gpd.SortedOutputFile method),
assign_handle() (seqtools.format.sam.SAMStream	close() (seqtools.format.sam.BAMFile method), 13
method), 16	close() (seqtools.format.sam.BGZF method), 14
average() (in module seqtools.statistics), 39	close() (seqtools.format.sam.SamtoolsBAMStream
areas growth and are processed and areas are areas areas are areas	method), 16
В	close() (seqtools.stream.GZippedOutputFile method), 39
BAM (class in seqtools.format.sam), 12	cmp() (seqtools.range.GenomicRange method), 32
BAMFile (class in sequools.format.sam), 13	cmp() (seqtools.structure.Junction method), 40
BAMIndex (class in seqtools.format.bamindex), 6	combine_context_errors() (seq-
BAMIndexRandomAccessPrimary (class in seq-	tools.errors.ErrorProfileFactory method),
tools.format.bamindex), 7	27
cools.formac.ounimacx), /	connected_nodes() (seqtools.graph.Graph method), 29

consecutive_exon_count()	(seq-	E
tools.structure.Transcript.ExonOverlap		Edge (class in seqtools.graph), 28
method), 42		emit_transcript() (seqtools.simulation.emitter.TranscriptomeEmitter
construct_cigar() (seqtools.align.Alignment method	), 20	method), 17
copy() (seqtools.format.fastq.FASTQ method), 10		encode_name() (in module seqtools.sequence), 38
copy() (seqtools.range.Bed method), 32		equals() (seqtools.range.GenomicRange method), 33
copy() (seqtools.range.GenomicRange method), 33		equals() (seqtools.structure.Junction method), 41
copy() (seqtools.sequence.Seq method), 37 copy() (seqtools.structure.Transcript method), 43		ErrorProfileFactory (class in seqtools.errors), 27
copy() (sequoois.structure.franscript method), 43		Exon (class in sequools.structure), 40
D		exon_overlap() (seqtools.structure.Transcript method), 43
decode_name() (in module seqtools.sequence), 38		exons (seqtools.format.gpd.GPD attribute), 10
destroy() (seqtools.format.bamindex.BAM	Index	exons (seqtools.structure.Transcript attribute), 44 external_cmd() (in module seq-
method), 6	mucx	external_cmd() (in module seq- tools.cli.utilities.bam_bgzf_index), 3
destroy() (seqtools.format.bamindex.BAMIndexRan	domA	
method), 7		tools.cli.utilities.bam_to_bed_depth), 4
different_random_nt()	(seq-	external_cmd() (in module seq-
tools.simulation.randomsource.RandomSo	_	tools.cli.utilities.fasta_to_fake_fastq), 4
method), 19		external_cmd() (in module seq-
distance() (seqtools.range.GenomicRange method),	33	tools.cli.utilities.fasta_to_tsv), 4
do_args() (in module seqtools.cli.cli_front), 5		external_cmd() (in module seq-
do_chunk() (in module	seq-	tools.cli.utilities.fastq_to_fasta), 4
tools.cli.utilities.bam_bgzf_index), 3		external_cmd() (in module seq-
do_inputs() (in module	seq-	tools.cli.utilities.fastq_to_tsv), 5
tools.cli.utilities.bam_bgzf_index), 3		external_cmd() (in module seq-
do_inputs() (in module	seq-	tools.cli.utilities.trim_fasta), 5
tools.cli.utilities.bam_to_bed_depth), 4		external_cmd() (in module seq-
do_inputs() (in module	seq-	tools.cli.utilities.trim_fastq), 5
tools.cli.utilities.fasta_to_fake_fastq), 4	torr	external_cmd() (in module seq-
do_inputs() (in module seqtools.cli.utilities.fasta_to 4	_tsv),	tools.cli.utilities.tsv_to_fasta), 5 external cmd() (in module seq-
do_inputs() (in module	seq-	external_cmd() (in module seq- tools.cli.utilities.tsv_to_fastq), 5
tools.cli.utilities.fastq_to_fasta), 4	seq	tools.cii.utiities.tsv_to_tastq), 5
do_inputs() (in module seqtools.cli.utilities.fastq_to	tsv).	F
5	/,	FASTA (class in seqtools.format.fasta), 9
do_inputs() (in module seqtools.cli.utilities.trim_fas	ta), 5	FASTA() (segtools.format.fasta.FASTA method), 9
do_inputs() (in module seqtools.cli.utilities.trim_fas		fasta() (seqtools.sequence.Seq method), 37
do_inputs() (in module seqtools.cli.utilities.tsv_to_f	asta),	FASTAData (class in seqtools.format.fasta), 9
5		FASTAFile (class in seqtools.format.fasta), 9
do_inputs() (in module seqtools.cli.utilities.tsv_to_f 5	astq),	FASTAFile.Chromosome (class in seqtools.format.fasta), 10
do_output() (in module	seq-	FASTAStream (class in seqtools.format.fasta), 10
tools.cli.utilities.bam_to_bed_depth), 4		FASTQ (class in seqtools.format.fastq), 10
dump_serialized() (seqtools.range.GenomicF	Range	FASTQ() (seqtools.format.fastq.FASTQ method), 10
method), 33		FASTQStream (class in seqtools.format.fastq), 10
dump_serialized() (seqtools.structure.Exon method) dump_serialized() (seqtools.structure.Junction method)		fetch_by_coord() (seqtools.format.sam.BAMFile method), 13
41 dump_serialized() (seqtools.structure.Transcript met	thod),	fetch_starting_at_coord() (seqtools.format.sam.BAMFile method), 13
43		find_cycle() (seqtools.graph.Graph method), 30
dump_serialized() (seqtools.structure.Transcrip method), 48	otome	G
		gauss() (seqtools.simulation.randomsource.RandomSource

method), 19

gc_content() (seqtools.sequence.Seq method), 38 GeneralErrorStats (class in seqtools.errors), 28 GenomicRange (class in seqtools.range), 32	get_chrom() (seqtools.structure.Transcript method), 44 get_cigar() (seqtools.format.sam.BAM method), 12 get_cigar() (seqtools.format.sam.SAM method), 15
get() (seqtools.cli.utilities.bam_bgzf_index.Queue method), 3	get_cigar() (seqtools.format.sam.SAM.PrivateValues method), 14
get_actual_original_query_range() (seq-	get_context_query_errors() (seq-
tools.format.sam.SAM method), 15	tools.errors.AlignmentErrors method), 23
get_actual_query_range() (seqtools.align.Alignment	get_context_target_errors() (seq-
method), 20	tools.errors.AlignmentErrors method), 23
get_adjusted_error_count() (seqtools.errors.BaseError	get_coord() (seqtools.format.sam.BAM method), 13
method), 25	get_coord_line_number() (seq-
get_after_probability() (seq-	tools.format.bamindex.BAMIndex method),
tools.errors.BaseError.UnobservableError	6
method), 25	get_coords_by_name() (seq-
get_after_type() (seqtools.errors.BaseError.UnobservableE method), 25	6
get_aligned_bases_count() (seqtools.align.Alignment method), 20	get_cut() (seqtools.simulation.permute.MakeCuts method), 18
get_alignment_errors() (seq-	get_depth_per_transcript() (seq-
tools.errors.ErrorProfileFactory method),	tools.structure.TranscriptLoci method), 48
27	get_directed_paths_from_node() (seqtools.graph.Graph
get_alignment_ranges() (seqtools.align.Alignment	method), 30
method), 20	get_direction() (seqtools.range.GenomicRange method),
get_alignment_ranges() (seqtools.format.sam.BAM method), 12	get_edges() (seqtools.graph.Graph method), 30
get_alignment_strings() (seqtools.align.Alignment	get_entry() (seqtools.format.fasta.FASTAStream
method), 20	method), 10
get_attributable_length() (seq-	get_entry() (seqtools.format.fastq.FASTQStream
tools. errors. Base Error. Observable Error	method), 10
method), 24	get_entry() (seqtools.format.psl.PSL.PrivateValues
get_attributable_length() (seq-	method), 11
tools.errors. Base Error. Unobservable Error	get_entry() (seqtools.format.sam.SAM.PrivateValues
method), 25	method), 14
get_base() (seqtools.errors.BaseError method), 25	get_error_probability() (seqtools.errors.BaseError
get_bed_array() (seqtools.range.GenomicRange method),	method), 25
33	get_error_probability() (seq-
get_bed_coordinates() (seqtools.range.GenomicRange method), 33	tools.errors.BaseError.ObservableError
get_bed_line() (seqtools.format.bed.Bed12 method), 8	method), 24 get_error_probability() (seq-
get_before_probability() (seq-	get_error_probability() (seq- tools.errors.BaseError.UnobservableError
tools.errors.BaseError.UnobservableError	method), 25
method), 25	get_exon() (seqtools.errors.AlignmentErrors.HPAGroup
get_before_type() (seq-	method), 22
tools.errors.BaseError.UnobservableError	get_exon_count() (seqtools.structure.Transcript method),
method), 25	44
<pre>get_block_bounds() (in module seqtools.format.bgzf), 8</pre>	get_exon_rules() (seqtools.structure.TranscriptLociMergeRule
<pre>get_block_start() (seqtools.format.bgzf.reader method), 8</pre>	method), 48
get_block_start() (seqtools.format.sam.BAM method), 12	get_fake_gpd_line() (seqtools.structure.Transcript
<pre>get_block_start() (seqtools.format.sam.BGZF method),</pre>	method), 44
14	get_fake_psl_line() (seqtools.structure.Transcript
get_changed_length() (seq-	method), 44
tools.errors.BaseError.ObservableError	get_file_position_string() (seqtools.format.sam.BAM
method), 24 get_children() (seqtools.graph.Graph method), 30	method), 13 get_filename() (seqtools.format.sam.BAM method), 13
Sec_emister() (bequotis.Stupit.Otupit filetilou), 50	501_Inchaine() (bequotis.formatisami.b) in inchiod), 13

get_gene_name() (seqtools.structure.Transcript method),	<b>C</b> =
44 get general errors() (segteels errors AlignmentErrors	tools.errors.ErrorProfileFactory method),
get_general_errors() (seqtools.errors.AlignmentErrors method), 23	get_names() (seqtools.format.bamindex.BAMIndex
get_genomic_coordinates() (seq-	method), 7
tools.range.GenomicRange method), 33	get_node1() (seqtools.graph.Edge method), 29
get_gpd_line() (seqtools.format.gpd.GPD method), 10	get_node2() (seqtools.graph.Edge method), 29
get_gpd_line() (seqtools.structure.Transcript method), 44	get_node_edges() (seqtools.graph.Graph method), 30
get_header() (seqtools.format.sam.BAMFile method), 13	get_node_ids() (seqtools.graph.Edge method), 29
get_header() (seqtools.format.sam.SAMStream method),	get_nodes() (seqtools.graph.Graph method), 30
16	get_nt() (seqtools.errors.AlignmentErrors.HPAGroup
<pre>get_homopolymer() (seqtools.errors.BaseError method),</pre>	method), 23
26	get_observable() (seqtools.errors.BaseError method), 26
get_homopolymer() (seq-	get_observable_error_probability() (seq-
tools.errors. Base Error. Observable Error	tools.errors.BaseError method), 26
method), 24	get_original_query_length() (seqtools.format.sam.SAM
get_HPAGroups() (seqtools.errors.AlignmentErrors	method), 15
method), 23	get_output() (in module seq-
get_id() (seqtools.graph.Edge method), 29	tools.cli.utilities.bam_to_bed_depth), 4
get_id() (seqtools.graph.Node method), 31	get_payload() (seqtools.graph.Node method), 31
get_id() (seqtools.structure.Transcript method), 45	get_payload() (seqtools.range.GenomicRange method),
get_index_line() (seqtools.format.bamindex.BAMIndex	33
method), 6 get_inner_start() (seqtools.format.bgzf.reader method), 8	get_payload() (seqtools.structure.Transcript method), 45 get_PSL() (seqtools.align.Alignment method), 20
get_inner_start() (seqtools.format.sam.BAM method), 13	get_PSL() (seqtools.format.psl.PSL method), 11
get_inner_start() (seqtools.format.sam.BGZF method),	get_quality() (seqtools.errors.AlignmentErrors.HPAGroup
14	method), 23
get_junction() (seqtools.structure.TranscriptGroup.Junction	
method), 47	tools.errors.AlignmentErrors method), 23
get_junction_string() (seqtools.structure.Transcript	get_query() (seqtools.errors.AlignmentErrors.HPAGroup
method), 45	method), 23
get_junctions_string() (seqtools.structure.Transcript	$get\_query\_base()  (seqtools.errors. BaseError. ObservableError$
method), 45	method), 24
get_juntol() (seqtools.structure.TranscriptLociMergeRules	
method), 48	tools.errors.ErrorProfileFactory method),
get_left_exon() (seqtools.structure.Junction method), 41	27
get_length() (seqtools.errors.AlignmentErrors.HPAGroup	get_query_context_errors() (seq-
method), 23 get_length() (seqtools.format.bamindex.BAMIndex	tools.errors.ErrorProfileFactory method),
method), 6	get_query_error() (seqtools.errors.AlignmentErrors
get_length() (seqtools.structure.Exon method), 40	method), 23
get_length() (seqtools.structure.Transcript method), 45	get_query_errors() (seqtools.errors.AlignmentErrors
get_line() (seqtools.format.bed.Bed12 method), 8	method), 24
get_line() (seqtools.format.gpd.GPD method), 10	get_query_length() (seqtools.align.Alignment method),
get_line() (seqtools.format.psl.PSL method), 11	21
get_line() (seqtools.format.sam.SAM method), 15	get_query_length() (seqtools.format.psl.PSL method), 11
get_line_number() (seqtools.format.sam.BAM method),	get_query_length() (seqtools.format.sam.SAM method),
13	15
get_longest_target_alignment_coords_by_name() (seq-	get_query_quality() (seqtools.align.Alignment method),
tools.format.bamindex.BAMIndex method),	21
6	get_query_quality() (seqtools.format.psl.PSL method),
get_merge_type() (seq- tools.structure.TranscriptLociMergeRules	get_query_quality() (seqtools.format.sam.SAM method),
method), 48	get_query_quanty() (seqtools.format.sam.sawi method),
	-

get_query_sequence() (seqtools.align.Alignment method), 21	get_strand() (seqtools.format.psl.PSL method), 12 get_strand() (seqtools.format.sam.SAM method), 15
get_query_sequence() (seqtools.errors.AlignmentErrors	get_strand() (seqtools.structure.Transcript method), 45
method), 24	get_string() (seqtools.errors.AlignmentErrors.HPAGroup method), 23
get_query_sequence() (seqtools.format.psl.PSL method),	get_string() (seqtools.errors.BaseError method), 26
gat guary saguance() (sagtools format sam SAM	
get_query_sequence() (seqtools.format.sam.SAM	get_string() (seqtools.errors.ErrorProfileFactory method),
method), 15	28
get_random_coord() (seq-	get_string() (seqtools.errors.GeneralErrorStats method),
tools.format.bamindex.BAMIndexRandomAcces	
method), 7	get_string() (seqtools.structure.Junction method), 41
get_range() (seqtools.format.gpd.GPD method), 10	get_tag() (seqtools.format.sam.BAM method), 13
get_range() (seqtools.format.sam.SAM method), 15	get_tag() (seqtools.format.sam.SAM method), 15
get_range() (seqtools.range.GenomicRange method), 34	get_tags() (seqtools.format.sam.SAM method), 16
get_range() (seqtools.structure.Exon method), 40	get_tags() (seqtools.format.sam.SAM.PrivateValues
get_range() (seqtools.structure.Transcript method), 45	method), 14
get_range() (seqtools.structure.TranscriptLoci method), 48	get_target() (seqtools.errors.AlignmentErrors.HPAGroup method), 23
get_range_start_coord() (seq- tools.format.bamindex.BAMIndex method),	get_target_base() (seqtools.errors.BaseError.ObservableError method), 25
7	get_target_context_error_report() (seq-
get_range_start_line_number() (seq-	tools.errors.ErrorProfileFactory method),
tools.format.bamindex.BAMIndex method),	28
7	get_target_context_errors() (seq-
get_range_string() (seqtools.range.GenomicRange method), 34	tools.errors.ErrorProfileFactory method), 28
get_range_string() (seqtools.structure.Junction method), 41	get_target_error() (seqtools.errors.AlignmentErrors method), 24
get_reference() (seqtools.align.Alignment method), 21 get_reference() (seqtools.format.psl.PSL method), 12	get_target_errors() (seqtools.errors.AlignmentErrors method), 24
get_report() (seqtools.errors.GeneralErrorStats method),	get_target_length() (seqtools.align.Alignment method),
get_report() (seqtools.graph.Graph method), 30 get_right_exon() (seqtools.structure.Junction method), 41	get_target_length() (seqtools.format.sam.BAM method),
get_roots() (seqtools.structure.struction method), 41	get_target_length() (seqtools.format.sam.SAM method),
get_SAM() (seqtools.align.Alignment method), 20	16
get_SAM() (seqtools.format.sam.SAM method), 14	get_target_range() (seqtools.align.Alignment method), 21
get_sequence() (seqtools.format.fasta.FASTAData method), 9	get_target_range() (seqtools.format.sam.SAM method), 16
get_sequence() (seqtools.format.fasta.FASTAFile method), 10	get_target_sequence() (seqtools.errors.AlignmentErrors method), 24
get_sequence() (seqtools.structure.Transcript method), 45	get_target_transcript() (seqtools.align.Alignment
get_sequence_length() (seqtools.format.sam.SAMHeader	method), 21
method), 16	get_transcript() (seqtools.structure.TranscriptGroup
get_sequence_lengths() (seq-	method), 47
tools.format.sam.SAMHeader method),	get_transcript_name() (seqtools.structure.Transcript method), 46
get_sequence_names() (seq-	get_transcripts() (seqtools.structure.TranscriptLoci
tools.format.sam.SAMHeader method),	method), 48
16	get_transcripts() (seqtools.structure.Transcriptome
get_stats() (seqtools.errors.GeneralErrorStats method),	method), 49
get_status_string() (seqtools.graph.Graph method), 31	get_type() (seqtools.errors.BaseError.ObservableError method), 25
get_status_string() (seqtools.graph.Graph method), 31 get_strand() (seqtools.align.Alignment method), 21	
501_511anu() (50410015.angn.Angninont inculou), 41	get_unaligned_lines() (seq-

tools.format.bamindex.BAMIndex method), 7	is_subset() (seqtools.structure.Transcript.JunctionOverlap method), 43
get_unaligned_start_coord() (seq-	1
tools.format.bamindex.BAMIndex method),	J
get_unobservable() (seqtools.errors.BaseError method),	Junction (class in seqtools.structure), 40 junction_overlap() (seqtools.structure.Transcript method), 46
get_unobservable_error_probability() (seq-	junctions (seqtools.format.gpd.GPD attribute), 10
tools.errors.BaseError method), 26 get_use_junctions() (seq-	junctions (seqtools.structure.Transcript attribute), 46
tools.structure.TranscriptLociMergeRules	K
method), 48	
get_use_multi_exons() (seq-	keys() (seqtools.format.fasta.FASTAData method), 9
tools.structure.TranscriptLociMergeRules method), 48	L
get_use_single_exons() (seq-	length() (seqtools.range.GenomicRange method), 34
tools.structure.TranscriptLociMergeRules method), 48	load_serialized() (seqtools.range.GenomicRange method), 34
get_weight() (seqtools.graph.Edge method), 29	load_serialized() (seqtools.structure.Exon method), 40
get_weighted_random_index() (seq-	load_serialized() (seqtools.structure.Junction method), 41
tools.simulation.randomsource.RandomSource	load_serialized() (seqtools.structure.Transcript method), 46
method), 19	load_serialized() (seqtools.structure.Transcriptome
GPD (class in seqtools.format.gpd), 10	method), 49
GPDStream (class in seqtools.format.gpd), 11 Graph (class in seqtools.graph), 29	Loci (class in seqtools.range), 35
GZippedOutputFile (class in seqtools.stream), 39	Locus (class in seqtools.range), 35
	LocusStream (class in seqtools.stream), 39
Н	M
has_quality() (seqtools.errors.AlignmentErrors method),	main() (in module seqtools.cli.cli_front), 5
24	main() (in module sequods.cli.cli_front), 3 main() (in module sequods.cli.utilities.bam_bgzf_index),
has_quality() (seqtools.errors.AlignmentErrors.HPAGroup method), 23	3
	main() (in module seq- tools.cli.utilities.bam_to_bed_depth), 4
I	main() (in module seq-
is_aligned() (seqtools.format.sam.SAM method), 16	tools.cli.utilities.fasta_to_fake_fastq), 4
is_any_error() (seqtools.errors.BaseError method), 26	main() (in module seqtools.cli.utilities.fasta_to_tsv), 4
is_bgzf() (in module seqtools.format.bgzf), 8	
is_compatible() (seqtools.structure.Transcript.ExonOverlap method), 42	main() (in module seqtools.cli.utilities.fastq_to_tsv), 5
is_compatible() (seqtools.structure.Transcript.JunctionOver	
method), 43	fiffain() (in module sequences clientilities true to foste), 5
is_directionless() (seqtools.graph.Edge method), 29	main() (in module seqtools.cli.utilities.tsv_to_fasta), 5 main() (in module seqtools.cli.utilities.tsv_to_fastq), 5
is_entry_key() (seqtools.format.psl.PSL.PrivateValues	MakeCuts (class in seqtools.simulation.permute), 17
method), 11	MakeErrors (class in seqtools.simulation.permute), 18
is_entry_key() (seqtools.format.sam.SAM.PrivateValues method), 14	match_exon_count() (seq-
is_full_overlap() (seqtools.structure.Transcript.ExonOverla	tools.structure.Transcript.ExonOverlap
method), 42	incurou), 43
is_full_overlap() (seqtools.structure.Transcript.JunctionOve	match_junction_count() (seq-
method), 43	tools.structure.Transcript.JunctionOverlap method), 43
is_header() (in module seqtools.format.sam), 17	median() (in module seqtools.statistics), 39
is_junction_line() (in module seqtools.format.sam), 17	merge() (seqtools.range.GenomicRange method), 34
is_subset() (seqtools.structure.Transcript.ExonOverlap method), 42	merge_cycles() (seqtools.graph.Graph method), 31

merge_down_loci() (seqtools.range.Loci method), 35 merge_ranges() (in module seqtools.range), 36	random_flip() (seqtools.simulation.permute.MakeErrors method), 18
MultiLocusStream (class in seqtools.stream), 40	random_insertion() (seq-
N	tools.simulation.permute.MakeErrors method), 18
N50() (in module seqtools.statistics), 38 n_count() (seqtools.sequence.Seq method), 38	random_nt() (seqtools.simulation.randomsource.RandomSource method), 19
next() (seqtools.format.fasta.FASTAStream method), 10	random_substitution() (seq-
next() (seqtools.format.fastq.FASTQStream method), 10	tools.simulation.permute.MakeErrors method),
next() (seqtools.format.gpd.GPDStream method), 11	RandomSource (class in seq-
next() (seqtools.format.sam.BAMFile method), 14	RandomSource (class in seq- tools.simulation.randomsource), 19
next() (seqtools.format.sam.SAMStream method), 16 next() (seqtools.range.BedArrayStream method), 32	ranges_to_coverage() (in module seqtools.range), 36
next() (seqtools.range.BedStream method), 32	rate_to_phred33() (in module seq-
next() (seqtools.stream.LocusStream method), 39	tools.simulation.permute), 19
next() (seqtools.stream.MultiLocusStream method), 40	rc() (in module seqtools.sequence), 38
Node (class in seqtools.graph), 31	rc() (seqtools.format.fastq.FASTQ method), 10
•	rc() (seqtools.sequence.Seq method), 38
O	read() (seqtools.format.bgzf.reader method), 9
overlap_size() (seqtools.range.GenomicRange method), 34	read() (seqtools.format.sam.BGZF method), 14 read_entry() (seqtools.format.gpd.GPDStream method),
overlap_size() (seqtools.structure.Transcript method), 46	read entry() (coatcols formet com PAMEIIs method) 14
overlaps() (seqtools.range.GenomicRange method), 34	read_entry() (seqtools.format.sam.BAMFile method), 14 read_entry() (seqtools.format.sam.SAMStream method),
overlaps() (seqtools.structure.Junction method), 41	16
overlaps_with_padding() (seqtools.range.GenomicRange	read_entry() (seqtools.range.BedArrayStream method),
method), 34	32
P	read_entry() (seqtools.range.BedStream method), 32
pad_ranges() (in module seqtools.range), 36	read_entry() (seqtools.stream.LocusStream method), 39
partition_graph() (seqtools.graph.Graph method), 31	read_entry() (seqtools.stream.MultiLocusStream
partition_loci() (seqtools.structure.TranscriptLoci	method), 40 read_entry2() (seqtools.format.sam.BAMFile method),
method), 48	14
phred33_to_rate() (in module seq-	reader (class in seqtools.format.bgzf), 8
tools.simulation.permute), 19	remove() (seqtools.format.fasta.FASTAData method), 9
print_alignment() (seqtools.align.Alignment method), 21	remove_edge() (seqtools.graph.Graph method), 31
print_range() (seqtools.range.GenomicRange method), 35	remove_node() (seqtools.graph.Graph method), 31
PSL (class in seqtools.format.psl), 11	remove_transcript() (seqtools.structure.TranscriptLoci
PSL.PrivateValues (class in seqtools.format.psl), 11	method), 48
	S
Q	SAM (class in seqtools.format.sam), 14
Queue (class in seqtools.cli.utilities.bam_bgzf_index), 3	SAM (class in sequois.format.sam), 14 SAM.PrivateValues (class in sequois.format.sam), 14
D	SAMHeader (class in seqtools.format.sam), 16
R	SAMStream (class in segtools, format.sam), 16
randint() (seqtools.simulation.randomsource.RandomSource	<sup>ce</sup> SamtoolsBAMStream (class in seqtools.format.sam), 16
method), 19	seek() (seqtools.format.bgzf.reader method), 9
random() (seqtools.simulation.randomsource.RandomSour method), 19	
random_deletion() (seq-	Seq (class in sequence), 37
tools.simulation.permute.MakeErrors method),	sequences align (module), 49
18	seqtools.align (module), 20 seqtools.cli (module), 6
random_flip() (in module seqtools.simulation.permute),	seqtools.cli.cli_front (module), 5
19	segtools cli utilities (module) 5

seqtools.cli.utilities.bam_bgzf_index (module), 3	set_entry() (seqtools.format.sam.SAM.PrivateValues
seqtools.cli.utilities.bam_to_bed_depth (module), 4	method), 14
seqtools.cli.utilities.fasta_to_fake_fastq (module), 4	set_exon_left() (seqtools.structure.Junction method), 41
seqtools.cli.utilities.fasta_to_tsv (module), 4	set_exon_right() (seqtools.structure.Junction method), 41
seqtools.cli.utilities.fastq_to_fasta (module), 4	set_exons_and_junctions_from_ranges() (seq-
seqtools.cli.utilities.fastq_to_tsv (module), 4	tools.structure.Transcript method), 46
seqtools.cli.utilities.trim_fasta (module), 5	set_gene_name() (seqtools.structure.Transcript method),
seqtools.cli.utilities.trim_fastq (module), 5	46
seqtools.cli.utilities.tsv_to_fasta (module), 5	set_is_leftmost() (seqtools.structure.Exon method), 40
seqtools.cli.utilities.tsv_to_fastq (module), 5	set_is_rightmost() (seqtools.structure.Exon method), 40
seqtools.errors (module), 22	set_junction_only() (seqtools.format.sam.SAMStream
seqtools.format (module), 17	method), 16
seqtools.format.bamindex (module), 6	set_juntol() (seqtools.structure.TranscriptLociMergeRules
seqtools.format.bed (module), 8	method), 48
seqtools.format.bgzf (module), 8	set_left() (seqtools.structure.Junction method), 41
seqtools.format.fasta (module), 9	set_left_junc() (seqtools.structure.Exon method), 40
seqtools.format.fastq (module), 10	set_lr_cuts() (seqtools.simulation.permute.MakeCuts
seqtools.format.gpd (module), 10	method), 18
seqtools.format.psl (module), 11	set_merge_rules() (seqtools.structure.TranscriptLoci
seqtools.format.sam (module), 12	method), 48
seqtools.graph (module), 28	set_minimum_distance() (seqtools.range.Loci method),
seqtools.range (module), 31	35
seqtools.sequence (module), 37	set_modified_base() (seq-
seqtools.simulation (module), 20	tools.simulation.permute.MakeErrors method),
seqtools.simulation.emitter (module), 17	18
seqtools.simulation.permute (module), 17	set_observable() (seqtools.errors.BaseError method), 26
seqtools.simulation.randomsource (module), 19	set_observed_base() (seq-
seqtools.statistics (module), 38	tools.simulation.permute.MakeErrors method),
seqtools.stream (module), 39	18
seqtools.structure (module), 40	set_payload() (seqtools.graph.Node method), 31
set() (seqtools.errors.BaseError.ObservableError	set_payload() (seqtools.range.GenomicRange method),
method), 25	35
set_after() (seqtools.errors.BaseError.UnobservableError	set_payload() (seqtools.structure.Transcript method), 46
method), 25	set_query_sequence() (seqtools.align.Alignment
set_after_context() (seq-	method), 21
tools.simulation.permute.MakeErrors method),	set_range() (seqtools.structure.Transcript method), 46
18	set_reference() (seqtools.align.Alignment method), 21
set_before() (seqtools.errors.BaseError.UnobservableError	set_right() (seqtools.structure.Junction method), 42
method), 25	set_right_junc() (seqtools.structure.Exon method), 40
set_before_context() (seq-	set_sequence() (seqtools.structure.Transcript method), 46
tools.simulation.permute.MakeErrors method),	set_sr_cuts() (seqtools.simulation.permute.MakeCuts
18	method), 18
set_cigar() (seqtools.format.sam.SAM.PrivateValues	set_strand() (seqtools.structure.Transcript method), 46
method), 14	set_tags() (seqtools.format.sam.SAM.PrivateValues
set_custom() (seqtools.simulation.permute.MakeCuts	method), 14
method), 18	set_transcript_name() (seqtools.structure.Transcript
set_direction() (seqtools.range.GenomicRange method),	method), 47
35	set_unobserved_after() (seqtools.errors.BaseError
set_entries_dict() (seqtools.format.psl.PSL.PrivateValues	method), 26
method), 11	set_unobserved_before() (seqtools.errors.BaseError
set_entries_dict() (seqtools.format.sam.SAM.PrivateValues	method), 27
method), 14	set_use_direction() (seqtools.range.Loci method), 35
set_entry() (seqtools.format.psl.PSL.PrivateValues	set_use_direction() (seqtools.range.Locus method), 36
method) 11	set use junctions() (seg-

tools.structure.TranscriptLociMergeRules	W
method), 48 set_weight() (seqtools.graph.Edge method), 29 set_weights_by_dict() (seq-	write() (seqtools.format.bgzf.writer method), 9 write() (seqtools.format.gpd.SortedOutputFile method),
tools.simulation.emitter.TranscriptomeEmitter	write() (seqtools.stream.GZippedOutputFile method), 39
method), 17 setup_tempdir() (in module seq-	write_context_error_report() (seq- tools.errors.ErrorProfileFactory method),
tools.cli.utilities.bam_bgzf_index), 3 setup_tempdir() (in module seq-	28 write_index() (in module seqtools.format.bamindex), 7
tools.cli.utilities.bam_to_bed_depth), 4 smooth_gaps() (seqtools.structure.Transcript method), 47 sort_genomic_ranges() (in module seqtools.range), 36 sort_header() (in module seqtools.format.sam), 17 sort_ranges() (in module seqtools.range), 36 SortedOutputFile (class in seqtools.format.gpd), 11 standard_deviation() (in module seqtools.statistics), 39 string_to_genomic_range() (in module seqtools.range), 36	writer (class in seqtools.format.bgzf), 9
subset() (seqtools.structure.Transcript method), 47 subtract() (seqtools.range.GenomicRange method), 35 subtract_range_array() (in module seqtools.range), 37 subtract_ranges() (in module seqtools.range), 37	
Т	
Transcript (class in seqtools.structure), 42 Transcript.ExonOverlap (class in seqtools.structure), 42 Transcript.JunctionOverlap (class in seqtools.structure), 43	
TranscriptGroup (class in seqtools.structure), 47 TranscriptGroup.JunctionGroup (class in seqtools.structure), 47	
TranscriptLoci (class in seqtools.structure), 47 TranscriptLociMergeRules (class in seqtools.structure), 48	
Transcriptome (class in seqtools.structure), 48 TranscriptomeEmitter (class in seqtools.simulation.emitter), 17	
type() (seqtools.errors.AlignmentErrors.HPAGroup method), 23	
U	
union() (seqtools.range.GenomicRange method), 35 union() (seqtools.structure.Transcript method), 47 union_range_array() (in module seqtools.range), 37 update_loci() (seqtools.range.Loci method), 35	
V	
validate() (seqtools.structure.Transcript method), 47 value() (seqtools.format.bed.Bed12 method), 8 value() (seqtools.format.gpd.GPD method), 10 value() (seqtools.format.psl.PSL method), 12 value() (seqtools.format.sam.BAM method), 13 value() (seqtools.format.sam.SAM method), 16 variance() (in module seqtools.statistics), 39	