Guided Assember API

API Documentation

December 7, 2013

Contents

1 Package src

1.1 Modules

1.2 Variables

Name	Description
__package__	**Value:** `None`

2 Module src.aligner

2.1 Functions

partition(*read*, *k*)

Partition the read into k parts

Parameters

`read:` Is the read to partition

`k:` Number of parts for the partition

kEdit(*p*, *t*, *k*)

Find an approximate match of p in t with up to k edits

Parameters

`p:` query string

`t:` reference string

`k:` max number of edits

test()

2.2 Variables

Name	Description
__package__	**Value:** `'src'`

2.3 Class Aligner

2.3.1 Methods

__**init**__(*self*, *ref_str*, *target*)

Object used for alignment of a target object that can generate multiple reads related to target string. i.e. reads to be aligned

Parameters

`ref_str:` the reference string

`target:` the target string

align(*self*)

Find an approximate match of the read in the reference. Pulls reads from the target_reads class.

alter_bwt(*self*, *no_opt*)

Alters the BWT dynamically based on the updates pending in the queue created in the align() method.

Parameters

`no_opt:` boolean if true do no optimizations i.e use a static bwt & no dbwt

3 Module src.assembler

3.1 Functions

assemble(*reference*, *target*, *threshold*, *min_consensus*, *no_opt*)

Execute method that runs the Guided Assembler on a given reference and target string

Parameters

`reference:`	an object of class Reference – representing the reference string
`target:`	an object of class Target – representing the target string
`threshold:`	the minimum value per index required for the addition of index to a contig
`min_consensus:`	the minimum total fraction of indexes that have come to concesus

eval_acc(*target_seq*, *contigs*)

Evaluate the accuracy of the assembly of target reads

Parameters

`target_seq:`	The target the sequence
`contigs:`	The contiguous sequences that are found

randStrings(*n*, *corrupt*)

Generate a random reference string and a related target string

Parameters

`n:`	length of toy strings
`corrupt:`	Percentage of corrupted nt

main()

3.2 Variables

Name	Description
__package__	**Value:** `'src'`

4 Module src.bwt

4.1 Functions

test(*s*)

main()

4.2 Variables

Name	Description
__package__	**Value:** `'src'`

4.3 Class BWT

object
└ **src.bwt.BWT**

Known Subclasses: src.dynamic_bwt.dBWT

4.3.1 Methods

__init__(*self*, *seq*)

A bwt class with a few auxilliary data structures

Parameters
`seq`: The seq in question

Overrides: object.__init__

new(*self*, *seq*)

Create a new bwt, F, L and Tally arrays (All necessary attributes for FM index)

Parameters
`seq`: the insertion seq

rank_bwt(*self*)

Given BWT string bw, return parallel list of B-ranks. Also returns tots: map from character to # times it appears. Adapted from Prof. Ben Langmend's example code

first_col(*self*, *tots*)

Return map from character to the range of rows prefixed by the character. Adapted from Prof. Ben Langmend's example code

Parameters

`tots:` a list with the a mapping of each character to the number of times it appears in F

Return Value

the character and total list

get_seq(*self*)

Make T from BWT(T) Adapted from Prof. Ben Langmend's example code

Return Value

the original sequence given the bwt

suffixArray(*self*, *s*)

Create a suffix array from a string s Adapted from Prof. Ben Langmend's example code

Parameters

`s:` the sequence we will use to create the suffix array

Return Value

the actual suffix array

bwtViaSa(*self*, *seq*)

Given T, returns BWT(T) by way of the suffix array Adapted from Prof. Ben Langmend's example code

Parameters

`seq:` the sequence we want to build the bwt from

Return Value

the BWT as a list

get_lcp(*self*)

TODO: DM

get_bwt(*self*)

TODO: DM

build_lcp(*self*, *seq*)

Parameters

`seq:` the sequence we will get lcp values for

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.3.2 Properties

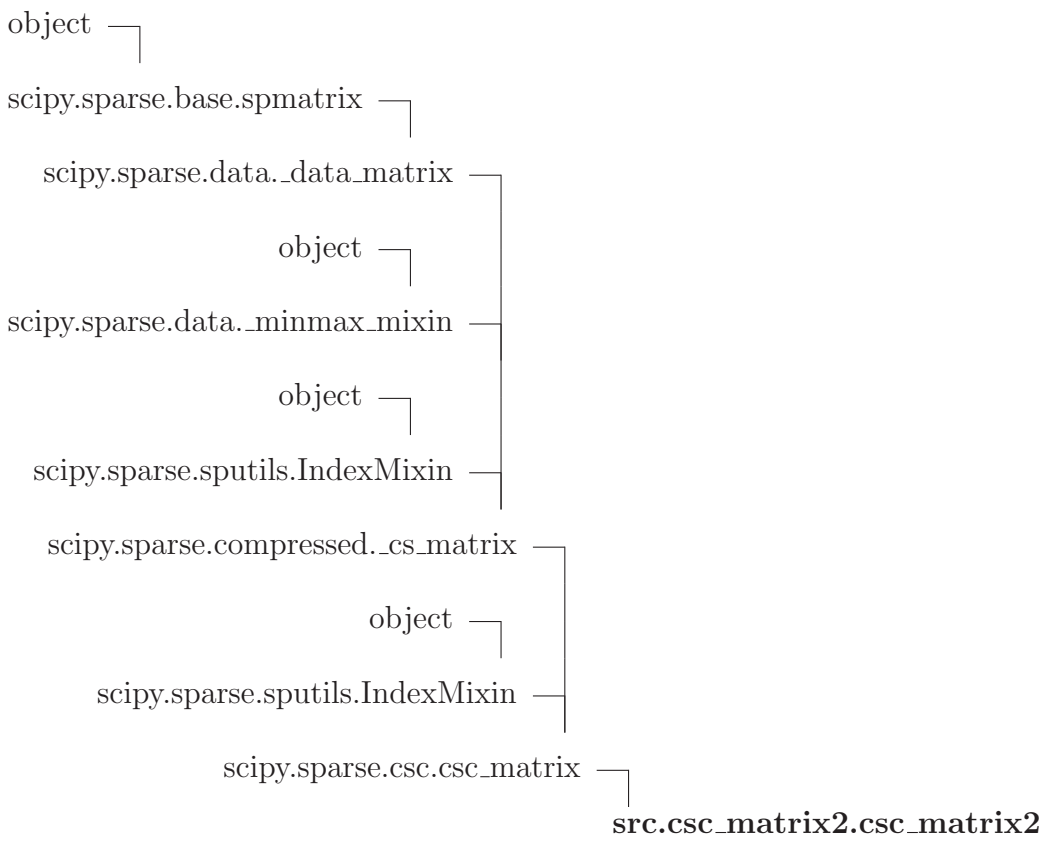
Name	Description
Inherited from object __class__	

5 Module src.csc_matrix2

5.1 Variables

Name	Description
__package__	**Value:** `'src'`

5.2 Class csc_matrix2



Sub-class of lil_matrix that allows permits popping rows off

5.2.1 Methods

pop_row(*self*)

append_col(*self*, *sp_mat*=`None`, *init*=`True`)

Add a column to the sparse matrix given another sp_mat to append. Used when adding a new letter to the alphabet.

Parameters

`sp_mat:` the sparse matrix to be appended to the self object

append_row(*self*)

Append a row to the bottom of a lil_matrix2 object

Inherited from scipy.sparse.csc.csc_matrix

__getitem__(), __iter__(), getcol(), getrow(), nonzero(), tocsc(), tocsr(), transpose()

Inherited from scipy.sparse.compressed._cs_matrix

__add__(), __eq__(), __ge__(), __gt__(), __init__(), __le__(), __lt__(), __ne__(), __radd__(), __rsub__(), __setitem__(), __sub__(), __truediv__(), check_format(), diagonal(), eliminate_zeros(), getnnz(), multiply(), prune(), sort_indices(), sorted_indices(), sum(), sum_duplicates(), toarray(), tocoo(), todia(), todok()

Inherited from scipy.sparse.data._data_matrix

__abs__(), __imul__(), __itruediv__(), __neg__(), arcsin(), arcsinh(), arctan(), arctanh(), astype(), ceil(), conj(), copy(), deg2rad(), expm1(), floor(), log1p(), rad2deg(), rint(), sign(), sin(), sinh(), sqrt(), tan(), tanh(), trunc()

Inherited from scipy.sparse.base.spmatrix

__bool__(), __div__(), __getattr__(), __iadd__(), __idiv__(), __isub__(), __len__(), __mul__(), __nonzero__(), __pow__(), __repr__(), __rmul__(), __str__(), asformat(), asfptype(), conjugate(), dot(), getH(), get_shape(), getformat(), getmaxprint(), mean(), reshape(), set_shape(), setdiag(), tobsr(), todense(), tolil()

Inherited from scipy.sparse.data._minmax_mixin

max(), min()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

5.2.2 Properties

Name	Description
Inherited from scipy.sparse.compressed._cs_matrix	

continued on next page

Name	Description
has_sorted_indices, nnz	
Inherited from scipy.sparse.data._data_matrix dtype	
Inherited from scipy.sparse.base.spmatrix shape	
Inherited from object __class__	

5.2.3 Class Variables

Name	Description
Inherited from scipy.sparse.base.spmatrix __array_priority__, ndim	

6 Module src.dynamic_bwt

6.1 Functions

test(*s*)

6.2 Variables

Name	Description
__package__	**Value:** `'src'`

6.3 Class dBWT

object
└
src.bwt.BWT
└
src.dynamic_bwt.dBWT

6.3.1 Methods

__init__(*self*, *seq*)

A dynamic Burrows Wheeler Transform class that supports on the fly changes to the seq that defined the BWT

Parameters

`seq`: str - the sequence you want to use to form the bwt

Overrides: object.__init__

build_psums(*self*)

Build the partial sums sparse matrix given a new seq

delete_one(*self*, *pos*)

Delete a char at some positin in the bwt

Parameters

`pos`: the position that should be deleted

replace_one(*self*, *char*, *pos*)

Replace a char at some position in the bwt

Parameters

`char`: the char that will replace the one at pos

`pos`: the positon that where the replacement is to occur

insert_one(*self*, *char*, *pos*)

Insert a character at a certain position 'pos' of the original sequence

Parameters

`char`: the character to insert

`pos`: the index of the original string where the character is to be inserted

updateSA_naive(*self*)

Update the suffix array to an alteration in the sequence defining the bwt

updateSA_dynamic(*self*)

reorder(*self*, *i*, *Lp*, *Fp*, *j*, *jp*, *psumsp*)

Move a row from row j to row jp

Parameters

`i`: the index (row) in the bwt where the change occured

`Lp`: the L' (prime) new L after the change

`Fp`: the F' (prime) new F after the change

`j`: the j actual position of j

`jp`: the j' (prime) expected position of j

`psumsp`: the partial sums' (prime)

moverow(*self*, *F*, *L*, *j*, *jp*, *psums*)

Take F and L and move row jp to j and moving others as necessary

Parameters

`F:` a list that corresponds to the first column of the bwm

`L:` a list that corresponds to the last column of the bwm

`j:` a row index i.e in range len(F/L)

`jp:` a row index i.e in range len(F/L)

Return Value

the new F and L with rows j & p switched

LF(*self*, *F*, *L*, *i*, *psums*)

LF computes a mapping from a char in F to a char in L in the BWM

Parameters

`F:` a list that corresponds to the first column of the bwm

`L:` a list that corresponds to the last column of the bwm

`i:` a row of the bwt

`psums:` a partial sums matrix

match(*self*, *seq*)

Parameters

`seq:` the sequence we are looking for

Return Value

an array of all perfect matches

get_rank(*self*, *row*, *char*, *psums*, *get_tot*=`True`)

" Get the rank of a letter at a particular row in the BWT.

Parameters

`row:` what row of the bwt to look at

`char:` the character whose rank we see

`psums:` the partial sums matrix corresponding to the state of the bwt we are concerned with

`get_tot:` boolean whether or not to get the total count for that character

Return Value

a rank and totals list of 2-item tuples

rank_bwt(*self*, *psums*)

Given BWT string bw, return parallel list of B-ranks. Also returns tots: map from character to # times it appears. Adapted from Prof. Ben Langmend's example code

Parameters

`psums`: the partial sums matrix corresponding to the state of the bwt we are concerned with

Overrides: src.bwt.BWT.rank_bwt

get_seq(*self*, *psums*)

Make T (The original sequence) from BWT(T) (The Burrows Wheeler Transform string)

Parameters

`psums`: the partial sums matrix corresponding to the state of the bwt we are concerned with

Return Value

the original sequence given the bwt

Overrides: src.bwt.BWT.get_seq

Inherited from src.bwt.BWT(Section 4.3)

build_lcp(), bwtViaSa(), first_col(), get_bwt(), get_lcp(), new(), suffixArray()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

6.3.2 Properties

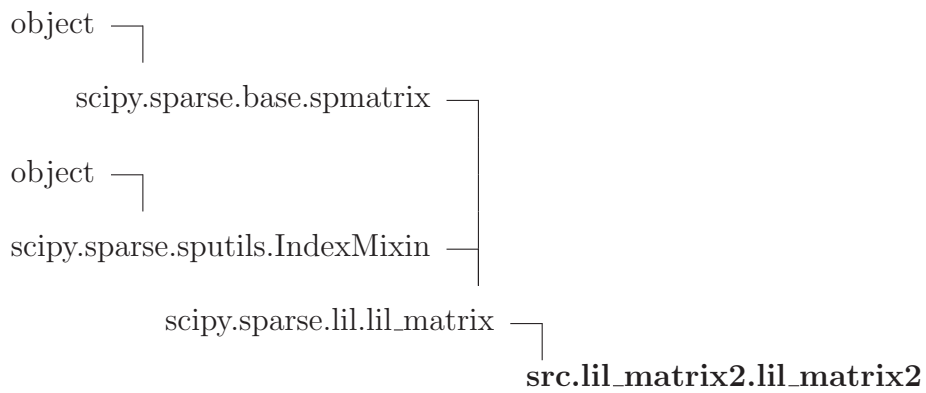
Name	Description
Inherited from object	
__class__	

7 Module src.lil_matrix2

7.1 Variables

Name	Description
__package__	**Value:** `'src'`

7.2 Class lil_matrix2



Sub-class of lil_matrix that allows permits popping rows off

7.2.1 Methods

pop_row(*self*)

append_col(*self*, *sp_mat*=`None`, *init*=`True`)

Add a column to the sparse matrix sp_mat1 Used when adding a new letter to the alphabet

Parameters

`sp_mat:` the sparse matrix to be appended to the self object

append_row(*self*)

Append a row to the bottom of a lil_matrix2 object

Inherited from scipy.sparse.lil.lil_matrix

__getitem__(), __iadd__(), __imul__(), __init__(), __isub__(), __itruediv__(), __setitem__(), __str__(), __truediv__(), copy(), getnnz(), getrow(), getrowview(), reshape(), set_shape(), toarray(), tocsc(), tocsr(), tolil(), transpose()

Inherited from scipy.sparse.base.spmatrix

__abs__(), __add__(), __bool__(), __div__(), __eq__(), __ge__(), __getattr__(), __gt__(), __idiv__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __neg__(), __nonzero__(), __pow__(), __radd__(), __repr__(), __rmul__(), __rsub__(), __sub__(), asformat(), asfptype(), astype(), conj(), conjugate(), diagonal(), dot(), getH(), get_shape(), getcol(), getformat(), getmaxprint(), mean(), multiply(), nonzero(), setdiag(), sum(), tobsr(), tocoo(), todense(), todia(), todok()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

7.2.2 Properties

Name	Description
Inherited from scipy.sparse.lil.lil_matrix	
nnz, shape	
Inherited from object	
__class__	

7.2.3 Class Variables

Name	Description
Inherited from scipy.sparse.base.spmatrix	
__array_priority__, ndim	

8 Module src.reference

8.1 Functions

test(*show*=`True`)

8.2 Variables

Name	Description
__package__	**Value:** `'src'`

8.3 Class Reference

object

src.reference.Reference

8.3.1 Methods

__init__(*self*, *R*, *data_counts*=`9`)

Object to hold a reference string and accompanying metadata associated with keeping track of how many matches/mismatches occur at each position

Parameters

`R:` Is the reference string

`data_counts:` the counts related to which letter landed at a particular index

Overrides: object.__init__

match(*self*, *idx*, *char*, *cnt*=`1`, *thresh*=`10`)

If there is a match in the reference at some position

Parameters

`idx:` the index in R where char matched

`char:` the char that matched

`cnt:` the number of times we should record char matched. Default=1

`thresh:` threshold to indicate a change in the refernce should be made

Return Value

True or False on (....something... TODO: SL)

build_hist(*self*, *coverage*, *show*=`False`, *save*=`False`, *save_fn*=`'max_hist_plot'`)

Build a histogram to determine what the maxes look & visualize match_count Might be used to determine a resonable threshold

Parameters

`coverage:` the average coverage for an single nt

`show:` Show visualization with match maxes

`save_fn:` Save to disk with this file name or else it will be the default

Return Value

the histogram array

get_contigs(*self*, *thresh*)

Use thresholding to determine what is a contig using the match_count

Parameters

`thresh:` anything under this value will not be included in the

Return Value

the contigs found in the string

plot_maxes(*self*, *show*=`False`)

Plot maxes to try to visualize where contigs will lie

Parameters

`show:` boolean on if you want to show the image on the screen

get_consensus(*self*, *thresh*)

Get the fraction of positions in the reference that have come to consus on the nt at that position.

Parameters

`thresh`: the number that defines what nt count is sufficient to be confident about the result at a single position

Return Value

number that gives the fraction of positions that have come to consensus

del_idx(*self*, *idx*)

Delete an index within the count array

Parameters

`idx`: the index we want deleted

insert_idx(*self*, *idx*)

Insert an index.

Parameters

`idx`: the index where we want to insert the charactere

zero_idx(*self*, *idx*)

Zero out an index within the count array

Parameters

`idx`: the index we want zeroed

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

8.3.2 Properties

Name	Description
Inherited from object	
__class__	

9 Module src.runtests

9.1 Functions

runtests(*num_tests*, *scriptname*, *out_fn*)

Run a series of tests to obtain benchmarks for guided assembler

Parameters

`num_tests:`	the number of tests to run
`scriptname:`	the name of the script containing the __main__ that we want to run
`out_fn:`	the name of the output file that we want to generate

9.2 Variables

Name	Description
__package__	**Value:** `'src'`

10 Module src.target_reads

10.1 Functions

pp(*var*)

Used in mapping operation for lists to ++ an index in the list

Parameters

`var`: some integer

Return Value

plus one to value of var

test()

main()

10.2 Variables

Name	Description
__package__	**Value:** `'src'`

10.3 Class Target

object

src.target_reads.Target

10.3.1 Methods

__init__(*self*, *p*, *read_length*, *T*, *seed*=`None`, *coverage*=`5`)

An Object that enables the partitioning &generation of synthetically mutated data given a full Target sequence read

Parameters

`p:`	the probability of having a polymorphism at any single letter within the read
`read_length:`	read lengths that we will produce
`T:`	the target string which we are trying to assemble using R, the ref
`seed:`	seed for random
`coverage:`	max redundancy at any single nt position

Overrides: object.__init__

get_read(*self*, ***kwargs*)

Get a single read from the Target sequence for a streaming read model

Parameters

`p:`	the probability of SNP occuring
`read_length:`	the read length
`max_trials:`	the maximum number of times to rand redraw the idx if we keep finding over-covered idxs

Return Value

a string with with a read obtained at a random position

mutate(*self*, *read*)

Use p to determine how to add SNPs to the returned string

Parameters

`read:` the read we want mutated

Return Value

a string with some SNPs added

get_read_list(*self*, ***kwargs*)

Return a list of even-coverage randomly sampled strings with possible SNPs given T. For the non-streaming version of the read splitting.

Parameters

`p:`	the probability of SNP occuring
`read_length:`	the read length
`save:`	boolean save or don't to disk
`save_fn:`	the filename you want to use to write to disk

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.3.2 Properties

Name	Description
Inherited from object	
__class__	

11 Module src.utils

11.1 Functions

Override(*interface_class*)

Method decorator for overriding class method name checking. Adapted from http://stackoverflow.com/questions/1167617/in-python-how-do-i-indicate-im-overriding-a-method

edta(*X*, *Y*)

Compute the Edit distance between two sequences

Parameters

`X`: any arbritrary string

`Y`: any arbritrary string

11.2 Variables

Name	Description
__package__	**Value:** `None`

12 Package tests

12.1 Modules

- **tdynamic_bwt** *(Section 13, p. 27)*

12.2 Variables

Name	Description
__package__	**Value:** `None`

13 Module tests.tdynamic_bwt

13.1 Functions

test_move_row()

Test the functionality of dbwt moverow

13.2 Variables

Name	Description
__package__	**Value:** `'tests'`

Index