EM2Tools

Release 0.1.0

Frédéric PLEWNIAK

Jun 16, 2020

CONTENTS:

CHAPTER

ONE

THE EM2TOOLS API REFERENCE

1.1 The "seq_record" module

Extension module to the Biopython Bio.SeqRecord module

class em2lib.seq_record.**SeqRecordEM2**(*seq*, ***kwargs*)

Extension to Biopython SeqRecord class

add_feature(***kwargs*)

Adds a feature to the current record according to arguments passed as **kwargs

Parameters **kwargs** – keyword arguments to pass to SeqFeatureEM2 class

feature_after(*position*, *strand=0*, *nearest=False*)

Retrieves the features immediately after (but not overlaping) the specified position, on one strand or both. If nearest is True, then only the nearest ones are returned.

Parameters

- **nearest** – if True, only the nearest features are returned. This only makes sense when strand is 0
- **strand** – strand specification of features to be returned. If strand is 0, then features on both strands are returned
- **position** – the position

Returns a list of features after the specified position

feature_before(*position*, *strand=0*, *nearest=False*)

Retrieves the features immediately before (but not overlaping) the specified position, on one strand or both. If nearest is True, then only the nearest ones are returned.

Parameters

- **nearest** – if True, only the nearest features are returned. This only makes sense when strand is 0
- **strand** – strand specification of features to be returned. If strand is 0, then features on both strands are returned
- **position** – the position

Returns a list of features before the specified position

join(*other=None*, *offset=0*, *keepself=True*)

Joins two SeqRecordEM2 objects into a new one representing the resulting merged sequence

Parameters

- **keepself** – if True and overlapping subsequences are different, then keep sequence from self record, otherwise keep the sequence of other record.
- **other** – the other SeqRecordEM2 object
- **offset** – the offset of the two sequences. If the value is negative, then the two sequences overlap.

Returns the result of merging records as a new SeqRecordEM2 object

overlap(*start*, *end=None*, *strand=0*)
Retrieves features that overlap a given position range.

Parameters

- **strand** – strand specification of features to be returned. If strand is 0, then features on both strands are returned. If feature.strand is 0, then all strands will match.
- **start** – start of range
- **end** – end of range, if None, then end=start

Returns a list of overlaping features

reverse_complement(*id=False*, *name=False*, *description=False*, *features=True*, *annotations=False*, *letter_annotations=True*, *dbxrefs=False*)
Reverse-complement the record adjusting features and their positions accordingly. The record id is conserved but if name is not specified 'reversed' is appended. All other arguments are passed and handled by the parent method. Note that the main goal for this method is to replace SeqRecord and Seq objects by their SeqRecordEM2 and SeqEM2 equivalents when reverse/complementing.

Parameters

- **id** – the id for the reversed record
- **name** – the name for the reversed record
- **description** – the description for the reversed record
- **features** – keep and adjust location of features if True
- **annotations** – keep annotations if True
- **letter_annotations** – keep letter_annotations if True
- **dbxrefs** – keep dbxrefs if True

Returns a reversed copy of the record

stitch(*other*, *fpos_in_self*, *fpos_in_other*, *feature_length*, *orientation=1*, ***kwargs*)
Stitches two records, that is, joins them according to an overlapping feature. The sequences may or may not overlap. If not, Ns or Xs are added to fill the gap. If they overlap, a warning is issued if sequences do not correspond exactly. The new record keeps track of the two original records as Features. By convention, the self record should contain the start position of the feature on the forward strand, the other contains the end position on the forward strand if orientation=1 or on the reverse strand if orientation=-1. If orientation is -1, then the other record is reversed/complemented before stitching and the position of the overlapping feature is modified accordingly. It is the user's responsibility to provide the records in the right order.

Parameters

- **other** – the other SeqRecordEM2 object to stitch
- **fpos_in_self** – feature position in self record (start position of feature)
- **fpos_in_other** – feature position in other record (end position of feature)
- **feature_length** – feature length

- **orientation** – the orientation of the other record relative to the self, either 1 if it is in the same orientation, -1 if other needs to be reversed before stitching, 0 if stranded but unknown, None for proteins
- **kwargs** – any additional parameters that may be passed to the constructor of the stitching feature in the new record

Returns the stitched record as a new SeqRecordEM2 object

surrounding_features(*position*, *strand=0*, *nearest=False*)
Retrieves all the features around a given position but not overlaping it. If nearest is True, then only the nearest features are returned.

Parameters

- **position** – the position
- **nearest** – if True, only the nearest features are returned.
- **strand** – strand specification of features to be returned. If strand is 0, then features on both strands are returned

Returns a list of features around the specified position

1.2 The "seq" module

Extension of Bio.Seq.Seq class from Biopython to add or improve functionalities

class em2lib.seq.**SeqEM2**(*data*, *seqtype*)
SeqEM2 class providing extension to Bio.Seq.Seq class of BioPython package.

classmethod dna(*data*)
Creates a DNA sequence

Parameters **data** – The sequence string

Returns a SeqEM2 DNA instance

is_protein()
Tests whether sequence was created as a protein

Returns

length_in_range(*minlength=None*, *maxlength=None*)
Checks whether the sequence length is with the specified range.

Parameters

- **minlength** – lower length bound
- **maxlength** – upper length bound

Returns True if sequence length is within specified range, False otherwise

classmethod protein(*data*)
Creates a protein sequence

Parameters **data** – The sequence string

Returns a SeqEM2 protein instance

re_search(*regex*)
Searches a sequence using a regular expression

Parameters **regex** – the regular expression

Returns a list of re.Match instances

search(*pattern*)
Searches sequence for a pattern specified with a fuzznuc or fuzzpro like syntax

Parameters **pattern** – the pattern to be searched for that is converted into a rgular expression

Returns a list of re.Match objects

1.3 The "seq_feature" module

Extension of Bio.SeqFeature module from Biopython to add or improve functionalities

class em2lib.seq_feature.**SeqFeatureEM2**(*parent=None*, *ref=None*, ***kwargs*)
SeqFeatureEM2 class providing extension to Bio.SeqFeature.SeqFeature class of BioPython package.

covers(*start*, *end*)
Determines whether feature covers the whole range specified by start and end

Parameters

- **start** – start of range either int or ExactPosition
- **end** – end of range either int or ExactPosition, if None then end=start

Returns True if feature covers the specified range

intersect(*other*, ***kwargs*)
Creates a new feature which is the intersection of feature and another one

Parameters **other** – the other feature

lies_within(*start*, *end*)

Determines whether feature lies entirely with the specified range. Fuzzy positions are turned into integers.

Parameters

- **start** – start of range either int or ExactPosition
- **end** – end of range either int or ExactPosition

Returns True if feature boundaries lie with the specified range.

move(*offset*)
Moves a feature by a certain offset

Parameters **offset** – offset by which the feature must be moved

overlaps(*start*, *end=None*)
Determines whether feature overlaps a position range.

Parameters

- **start** – start of range either int or ExactPosition
- **end** – end of range either int or ExactPosition

Returns True if feature overlaps range

1.4 The "seq_utils" module

Extension module to the Biopython Bio.SeqUtils module

class em2lib.seq_utils.**GFF**(*feature_list=None*, *input_df=None*)
Manipulation of features based upon gffpandas package

add_feature_list(*feature_list=None*)
Adds a list of feature to the list of an existing GFF object

Parameters **feature_list** – list of features to add to DataFrame

Returns the GFF object with feature list appended

static df_from_feature(*feature*)
Create a pandas DataFrame from a feature (SeqFeatureEM2 or SeqFeature)

Parameters **feature** – the feature to convert into a dataframe

Returns the resulting dataframe

to_feature_list(*parents=None*)
Converts features in a GFF object into a list of SeqFeatureEM2 objects

Parameters **parents** – list of references to parent SeqRecord objects or a single parent reference if all features are defined in the same parent. If it is a list, it should be of the same length as the dataframe, repeating references as needed to get the right number.

Returns a list of SeqFeatureEM2 objects

class em2lib.seq_utils.**SeqFilter**
A class for the creation of a sequence filter to specify filtering criteria and applying the filter to a list of sequence records. (minlength: minimum length of sequence, maxlength: maximum length of sequence, pattern: sequence pattern, name: sequence name, keep: boolean, if True, keep the records respecting the criteria, otherwise, discard them and keep the others.

apply(*records*)
Filters a list of SeqRecords instances, keeping only records satisfying the specified criteria of length, match of a pattern, name specification. It is possible to invert the filtering process by setting the keep boolean to False and thus only keep records which do not satisfy the criteria.

Parameters **records** – list of SeqRecord instances to apply

Returns the filtered list of records

keep(*keep=True*)
Boolean defining whether the matching sequences must be kept (True) or removed (False)

Parameters **keep** – True to keep positive sequences, False to remove them

Returns SeqFilter instance

length(*minlength=None*, *maxlength=None*)
Minimal and maximal length specification

Parameters

- **minlength** – minimal accepted length
- **maxlength** – maximal accepted length

Returns SeqFilter instance

length_applies(*rec*)
test whether length criterion applies to the sequence record

Parameters **rec** – the sequence record to test

Returns boolean True if criterion applies or False otherwise

name(*name=None*)
sequence record name specification

Parameters **name** – name regular expression

Returns SeqFilter instance

name_applies(*rec*)
test whether name criterion applies to the sequence record

Parameters **rec** – the sequence record to test

Returns boolean True if criterion applies or False otherwise

pattern(*pattern=None*)
pattern specification

Parameters **pattern** – pattern that must be in the sequence

Returns SeqFilter instance

pattern_applies(*rec*)
test whether parameter criterion applies to the sequence record

Parameters **rec** – the sequence record to test

Returns boolean True if criterion applies or False otherwise

em2lib.seq_utils.**ambiguous2string**(*code*, *protein=False*)
Converts an ambiguous residue into a string with all compatible unambiguous residues. If the input code is not ambiguous, it is returned without any conversion.

Parameters

- **code** – the input code to be converted into a list of residues.
- **protein** – True if residue is amino-acid

Returns a string corresponding to the unambiguous residues compatible with the input code

em2lib.seq_utils.**isambiguous**(*code*, *protein=False*)
Checks code is an ambiguous residue specification or not.

Parameters

- **code** – the input code that must be checked for ambiguity
- **protein** – True if code is amino-acid code

Returns Boolean True if code is ambiguous, False otherwise

em2lib.seq_utils.**pattern2regex**(*pattern*, *protein=False*)
Converts a fuzznuc or fuzzpro-like pattern into a regular expression that can be used to search a sequence string. [ABC] => any of ABC residues, {ABC} => any residue except ABC, <ABC... => start of sequence, ...ABC> => end of sequence, A(n)(ABC)(n) => repeat residue or subsequence n times, A(n,m)(ABC)(n,m) => repeat residue or subsequence from n up to m times.

Parameters

- **pattern** – the pattern definition (string)
- **protein** – True if pattern applies to a protein sequence, False otherwise.

Returns the regular expression pattern as a string

1.5 The "argparse_em2" module

Extension module to the standard argparse module. Adding custom actions and argument verification methods.

class em2lib.argparse_em2.**GetList**(*option_strings*, *dest*, *nargs='+'*, ***kwargs*)

An argparse custom action to return a list from an argument containing a list of elements and/or file names. Files are supposed to contain one element of the list per line. There can be more than one file and the argument may take a combination of elements and files. In all cases, the returned list will contain all the specified elements without any checking for redundancy. If you need a non redundant set instead of a list, then use GetSet action instead.

static arg2list(*values*)

This method converts the argument values containing elements and/or files containing elements into a list of elements.

Parameters **values** – argument values, this is supposed to be a list of arguments or None (returns an empty list)

Returns the list of elements or an empty list if the argument was None

class em2lib.argparse_em2.**GetSet**(*option_strings*, *dest*, *nargs='+'*, ***kwargs*)

An argparse custom action to return a set from an argument containing a list of elements and/or file names. Files are supposed to contain one element of the list per line. There can be more than one file and the argument may take a combination of elements and files. In all cases, the returned set will contain all the specified elements keeping only one copy of each element. If you do not want to remove redundancy, then use GetList action instead.

static arg2set(*values*)

This method converts the argument values containing elements and/or files containing elements into a set of elements.

Parameters **values** – argument values, this is supposed to be a list of arguments or None (returns an empty set)

Returns the set of elements or an empty set if the argument was None

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e

INDEX

O

P

R

S

T