fastinterval Documentation

Release 0.1.1

James Casbon

June 27, 2013

CONTENTS

CHAPTER

ONE

OVERVIEW

An simple interval class for DNA sequences from FASTA files that provides fast access to sequences and methods for interval logic on those sequences.

Usually you will create a *Genome* and then use that object to create intervals. The intervals have a sequence property that will look up the actual sequence:

```
>>> from fastinterval import Genome, Interval
>>> test_genome = Genome('test/example.fa')
>>> int1 = test_genome.interval(100, 150, chrom='1')
>>> print int1
1:100-150:
>>> print int1.sequence
GATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGA
```

fastinterval uses pyfasta to retrieve the sequence, so the access is mmapped (i.e fast). It supports strandedness, which will be respected when accessing the sequence:

```
>>> int2 = test_genome.interval(100, 150, chrom='1', strand=-1)
>>> print int2.sequence
TCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATC
```

The Interval class supports many interval operations:

```
>>> int1 = test_genome.interval(100, 150, chrom='1')
>>> int2 = test_genome.interval(125, 175, chrom='1')
>>> int1.distance(int2)
0
>>> int1.span(int2)
Interval(100, 175)
>>> int1.overlaps(int2)
True
>>> int1.is_contiguous(int2)
True
>>> int1 in int2
False
>>> int1.intersection(int2)
Interval(125, 150)
>>> int1.union(int2)
Interval(100, 175)
>>> Interval.merge([int1, int2, test_genome.interval(200,250, chrom='1')])
[Interval(100, 175), Interval(200, 250)]
```

The Interval class is also based on bx python intervals. So you can pass in a value attritbue to point to an external object, and create interval trees and so on.

```
>>> from bx.intervals.intersection import IntervalTree
>>> int3 = test_genome.interval(150, 200, chrom='1', value='foo')
>>> tree = IntervalTree()
>>> _ = map(tree.insert_interval, (int1, int2, int3))
>>> tree.find(190, 195)
[Interval(150, 200, value=foo)]
```

CHAPTER

TWO

INSTALLATION

fastinterval can be installed with pip:

```
pip install fastinterval
```

CHAPTER

THREE

DEVELOPMENT

Bugs, patches, etc should be submitted to the github repository: https://github.com/jamescasbon/fastinterval

CHAPTER

FOUR

API DOCUMENTATION

4.1 Interval

class `fastinterval.`**`Interval`**(*start*, *stop*, *genome=None*, ***kws*)

A genomic interval

`add_border`(*size=0*, *upstream=0*, *downstream=0*)

return interval with some bases added to each end

`copy`(***kws*)

Copy this interval, and optionally provide a dict of new attrs

`distance`(*other*)

return the distance between two intervals

classmethod `from_string`(*loc*, ***kws*)

Create an interval from a chrx:start-end style string

`intersection`(*other*)

Return the interval containing the intersection of two intervals

`is_contiguous`(*other*)

Return True if the intervals are overlapping or contiguous

classmethod `merge`(*intervals*, *merge_contiguous=False*, ***kwargs*)

merge a list of intervals and return a list of intervals

By default, the intervals must be overlapping to be merged. If you want to merge contiguous intervals, set merge_contiguous to True.

`overlaps`(*other*)

Return True if the intervals share at least one base

`sequence`

Return the DNA from this intecal as a string

`span`(*other*, ***kws*)

Return an interval spanning two intervals

`span_between`(*other*, ***kws*)

Return an Inteval spanning the gap between two intervals

`truncate`(*size*)

truncate this interval to size, respecting the orientation

`union`(*other*, *merge_contiguous=False*)

Return an interval containing the interval of two overlapping interval

4.2 Genome

class `fastinterval.`**`Genome`**(*fname*, **args*, ***kws*)

A convienience for creating intervals on the same genome

`from_string`(*data*)

docstring for from_string

`interval`(*start*, *end*, ***kws*)

return an interval on this genome

4.3 MinimalSpanningSet

class `fastinterval.`**`MinimalSpanningSet`**(*targets*, *candidates*, *score_function=None*, *sort_key=None*)

Create a minimal spanning set for target intervals from a set of candidates

CHAPTER

FIVE

INDICES AND TABLES

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

PYTHON MODULE INDEX

f

fastinterval, **??**