DNAStrand

Generated by Doxygen 1.8.18

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 DNAStrand Namespace Reference

Classes

- class DNAStrand

  Class that implements a simple model of a DNA Strand, with a sequence of characters 'A', 'G', 'C', 'T'.

Functions

- def main (args=None)

  Main program for testing.

4.1.1 Detailed Description

Playing with string matching.

Author

Paulo Roma (structure, class, methods and main), William Souza (implementations, besides main)

Since

15/12/2019

See also

https://www.sciencedirect.com/topics/medicine-and-dentistry/dna-strand

4.1.2 Function Documentation

4.1.2.1 main()

```
def DNAStrand.main (
              args = None )
```

Main program for testing.

Parameters

args	two DNA strands.

4.2 DNAStrandTest Namespace Reference

Classes

- class DNAStrandTest

  Class for testing certain aspects of the behavior of DNAStrand.

4.2.1 Detailed Description

Class for testing the DNAStrand matching.

Author

William Souza

Since

05/02/2020

Chapter 5

Class Documentation

5.1 DNAStrand.DNAStrand Class Reference

Class that implements a simple model of a DNA Strand, with a sequence of characters 'A', 'G', 'C', 'T'.

Public Member Functions

- def __init__ (self, givenData)

  Constructs a DNAStrand with the given string of data, normally consisting of characters 'A', 'C', 'G', and 'T'.
- def __str__ (self)

  Returns a string representing the strand data of this DNAStrand.
- def createComplement (self)

  Returns a new DNAStrand that is the complement of this one, that is, 'A' is replaced with 'T' and so on.
- def findMatchesWithLeftShift (self, other, shift)

  Returns a string showing which characters in this strand are matched with 'other', when shifted left by the given amount.
- def findMatchesWithRightShift (self, other, shift)

  Returns a string showing which characters in this strand are matched with 'other', when shifted right by the given amount.
- def findMaxPossibleMatches (self, other)

  Returns the maximum possible number of matching base pairs, when the given sequence is shifted left or right by any amount.
- def countMatchesWithLeftShift (self, other, shift)

  Returns the number of matching pairs, when 'other' is shifted to the left by 'shift' positions.
- def countMatchesWithRightShift (self, other, shift)

  Returns the number of matching pairs, when 'other' is shifted to the right by 'shift' positions.
- def isValid (self)

  Determines whether all characters in this strand are valid ('A', 'G', 'C', or 'T').
- def letterCount (self, ch)

  Counts the number of occurrences of the given character in this strand.
- def matches (self, c1, c2)

  Returns True if the two characters form a base pair ('A' with 'T' or 'C' with 'G').

Public Attributes

- strand

  Strand of this DNA, in upper case.

5.1.1 Detailed Description

Class that implements a simple model of a DNA Strand, with a sequence of characters 'A', 'G', 'C', 'T'.

'A' char corresponds to 'T', and 'C' char to 'G', those chars form a base pair each. All the methods within this class are used for the manipulation and study of the DNAStrands.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def DNAStrand.DNAStrand.__init__ (
              self,
              givenData )
```

Constructs a DNAStrand with the given string of data, normally consisting of characters 'A', 'C', 'G', and 'T'.

Raises a ValueError exception, in case of an invalid givenData strand.

Parameters

givenData	string of characters for this DNAStrand.

5.1.3 Member Function Documentation

5.1.3.1 countMatchesWithLeftShift()

```
def DNAStrand.DNAStrand.countMatchesWithLeftShift (
              self,
              other,
              shift )
```

Returns the number of matching pairs, when 'other' is shifted to the left by 'shift' positions.

Parameters

other	given DNAStrand to match with this strand.
shift	number of positions to shift other to the left.

Returns

number of matching pairs.

5.1.3.2 countMatchesWithRightShift()

```
def DNAStrand.DNAStrand.countMatchesWithRightShift (
              self,
              other,
              shift )
```

Returns the number of matching pairs, when 'other' is shifted to the right by 'shift' positions.

Parameters

other	given DNAStrand to be matched with this one.
shift	number of positions to shift other to the right.

Returns

number of matching pairs.

5.1.3.3 createComplement()

```
def DNAStrand.DNAStrand.createComplement (
              self )
```

Returns a new DNAStrand that is the complement of this one, that is, 'A' is replaced with 'T' and so on.

Returns

complement of this DNA.

5.1.3.4 findMatchesWithLeftShift()

```
def DNAStrand.DNAStrand.findMatchesWithLeftShift (
              self,
              other,
              shift )
```

Returns a string showing which characters in this strand are matched with 'other', when shifted left by the given amount.

Parameters

other	given DNAStrand.
shift	number of positions to shift other to the left.

Returns

a copy of this strand, where matched characters are upper case and unmatched, lower case.

5.1.3.5 findMatchesWithRightShift()

```
def DNAStrand.DNAStrand.findMatchesWithRightShift (
              self,
              other,
              shift )
```

Returns a string showing which characters in this strand are matched with 'other', when shifted right by the given amount.

Parameters

other	given DNAStrand.
shift	number of positions to shift other to the right.

Returns

a copy of this strand, where matched characters are upper case and unmatched, lower case.

5.1.3.6 findMaxPossibleMatches()

```
def DNAStrand.DNAStrand.findMaxPossibleMatches (
              self,
              other )
```

Returns the maximum possible number of matching base pairs, when the given sequence is shifted left or right by any amount.

Parameters

other	given DNAStrand to be matched with this one.

Returns

maximum number of matching pairs.

5.1.3.7 isValid()

```
def DNAStrand.DNAStrand.isValid (
              self )
```

Determines whether all characters in this strand are valid ('A', 'G', 'C', or 'T').

Returns

True if valid, and False otherwise.

5.1.3.8 letterCount()

```
def DNAStrand.DNAStrand.letterCount (
              self,
              ch )
```

Counts the number of occurrences of the given character in this strand.

Parameters

ch	given character.

Returns

number of occurrences of ch.

5.1.3.9 matches()

```
def DNAStrand.DNAStrand.matches (
              self,
              c1,
              c2 )
```

Returns True if the two characters form a base pair ('A' with 'T' or 'C' with 'G').

Parameters

c1	first character.
c2	second character.

Returns

True if they form a base pair, and False otherwise.

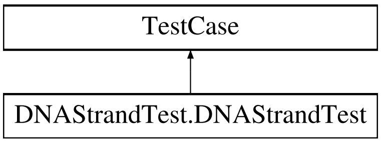
The documentation for this class was generated from the following file:

- DNAStrand.py

5.2 DNAStrandTest.DNAStrandTest Class Reference

Class for testing certain aspects of the behavior of DNAStrand.

Inheritance diagram for DNAStrandTest.DNAStrandTest:



Public Member Functions

- def test_isValid (self)

  Test for isValid method, should return a boolean indicating if the string given has only valid chars for the strand (true) ou not (false).
- def test_createComplement (self)

  Test for createComplement method, should return a string with the complement (all char matching with the given strand).
- def test_findMatchesWithLeftShift (self)

  Test for findMatchesWithLeftShift method, should return a string with the matching chars in uppercase and the non-matching in lowercase, when the 'other' strand is shifted left.
- def test_findMatchesWithRightShift (self)

  Test for findMatchesWithRightShift method, should return a string with the matching chars in uppercase and the non-matching in lowercase, when the 'other' strand is shifted right One relation between strands was tested, with different shift values, looking towards negative indexes, 0 index and positive.
- def test_findMaxPossibleMatches (self)

  Test for findMaxPossibleMatches method, should return an integer representing the maximum number of matches, independent of how many shifts are made with the strands.
- def test_countMatchesWithLeftShift (self)

  Test for countMatchesWithLeftShift method, should return an integer representing the number of matches between the strands, when other is shifted left.
- def test_countMatchesWithRightShift (self)

  Test for countMatchesWithRightShift method, should return an integer representing the number of matches between the strands, when other is shifted right.
- def test_letterCount (self)

  Test for letterCount method, should return an integer representing the number of times the given character appears in the strand.
- def test_matches (self)

  Test for matches method, should return true if characters given matches and false if not.

Static Public Attributes

- d1 = DNAStrand.DNAStrand('TCAT')

  Two DNAStrand objects to be used in the tests.
- d2 = DNAStrand.DNAStrand('AGAGCAT')

  DNAStrand object to be used in the tests.

5.2.1 Detailed Description

Class for testing certain aspects of the behavior of DNAStrand.

This class tests all DNAStrand methods with different inputs, and prints OK if the class passes all the tests.

5.2.2 Member Function Documentation

5.2.2.1 test_countMatchesWithLeftShift()

```
def DNAStrandTest.DNAStrandTest.test_countMatchesWithLeftShift (
              self )
```

Test for countMatchesWithLeftShift method, should return an integer representing the number of matches between the strands, when other is shifted left.

One relation between strands was tested, with different shift values, looking towards negative indexes, 0 index and positive.

5.2.2.2 test_countMatchesWithRightShift()

```
def DNAStrandTest.DNAStrandTest.test_countMatchesWithRightShift (
              self )
```

Test for countMatchesWithRightShift method, should return an integer representing the number of matches between the strands, when other is shifted right.

One relation between strands was tested, with different shift values, looking towards negative indexes, 0 index and positive.

5.2.2.3 test_createComplement()

```
def DNAStrandTest.DNAStrandTest.test_createComplement (
              self )
```

Test for createComplement method, should return a string with the complement (all char matching with the given strand).

Two strands where tested expecting correct complements.

5.2.2.4 test_findMatchesWithLeftShift()

```
def DNAStrandTest.DNAStrandTest.test_findMatchesWithLeftShift (
              self )
```

Test for findMatchesWithLeftShift method, should return a string with the matching chars in uppercase and the non-matching in lowercase, when the 'other' strand is shifted left.

One relation between strands was tested, with different shift values, looking towards negative indexes, 0 index and positive.

5.2.2.5 test_findMatchesWithRightShift()

```
def DNAStrandTest.DNAStrandTest.test_findMatchesWithRightShift (
              self )
```

Test for findMatchesWithRightShift method, should return a string with the matching chars in uppercase and the non-matching in lowercase, when the 'other' strand is shifted right One relation between strands was tested, with different shift values, looking towards negative indexes, 0 index and positive.

5.2.2.6 test_findMaxPossibleMatches()

```
def DNAStrandTest.DNAStrandTest.test_findMaxPossibleMatches (
              self )
```

Test for findMaxPossibleMatches method, should return an integer representing the maximum number of matches, independent of how many shifts are made with the strands.

One test was made, with the two strands given.

5.2.2.7 test_isValid()

```
def DNAStrandTest.DNAStrandTest.test_isValid (
              self )
```

Test for isValid method, should return a boolean indicating if the string given has only valid chars for the strand (true) ou not (false).

Two strands where tested, both with valid outputs.

5.2.2.8 test_letterCount()

```
def DNAStrandTest.DNAStrandTest.test_letterCount (
              self )
```

Test for letterCount method, should return an integer representing the number of times the given character appears in the strand.

Were tested uppercase, lowercase and invalid letters.

5.2.2.9 test_matches()

```
def DNAStrandTest.DNAStrandTest.test_matches (
              self )
```

Test for matches method, should return true if characters given matches and false if not.

Where tested uppercase, lowercase, valid and invalid characters.

The documentation for this class was generated from the following file:

- DNAStrandTest.py

Index