

DNAStrand

Generated by Doxygen 1.8.17

1 DNAStrand.py: AD1 Programação com Interfaces Gráficas 2020.1	1
1.1 Introduction	1
1.2 Software	1
1.3 References	1
2 Namespace Index	3
2.1 Packages	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 DNAStrand Namespace Reference	11
6.1.1 Function Documentation	11
6.1.1.1 main()	11
6.2 DNAStrandTest Namespace Reference	12
6.2.1 Detailed Description	12
6.2.2 Function Documentation	12
6.2.2.1 main()	12
7 Class Documentation	13
7.1 DNAStrand.DNAStrand Class Reference	13
7.1.1 Constructor & Destructor Documentation	14
7.1.1.1 __init__()	14
7.1.2 Member Function Documentation	14
7.1.2.1 __str__()	14
7.1.2.2 countMatchesWithLeftShift()	14
7.1.2.3 countMatchesWithRightShift()	15
7.1.2.4 createComplement()	15
7.1.2.5 findMatchesWithLeftShift()	15
7.1.2.6 findMatchesWithRightShift()	16
7.1.2.7 findMaxPossibleMatches()	16
7.1.2.8 isValid()	17
7.1.2.9 letterCount()	17
7.1.2.10 matches()	17
7.1.3 Member Data Documentation	18
7.1.3.1 strand	18
7.1.3.2 symbols	18

7.2 DNAStrandTest.DNAStrandClass Class Reference	18
7.2.1 Detailed Description	20
7.2.2 Member Function Documentation	20
7.2.2.1 setUp()	20
7.2.2.2 test_countMatchesWithLeftShift()	20
7.2.2.3 test_countMatchesWithRightShift()	20
7.2.2.4 test_createComplement()	20
7.2.2.5 test_findMatchesWithLeftShift()	21
7.2.2.6 test_findMatchesWithRightShift()	21
7.2.2.7 test_findMaxPossibleMatches()	21
7.2.2.8 test_isValid()	21
7.2.2.9 test_letterCount()	21
7.2.2.10 test_matches()	21
7.2.3 Member Data Documentation	22
7.2.3.1 a1	22
7.2.3.2 a1ls0	22
7.2.3.3 a1ls1	22
7.2.3.4 a1ls2	22
7.2.3.5 a1ls3	22
7.2.3.6 a1rs0	22
7.2.3.7 a1rs1	22
7.2.3.8 a1rs2	23
7.2.3.9 a1rs3	23
7.2.3.10 a1rs4	23
7.2.3.11 a1rs5	23
7.2.3.12 a1rs6	23
7.2.3.13 a2	23
7.2.3.14 a2ls0	23
7.2.3.15 a2ls1	23
7.2.3.16 a2ls2	24
7.2.3.17 a2ls3	24
7.2.3.18 a2ls4	24
7.2.3.19 a2ls5	24
7.2.3.20 a2ls6	24
7.2.3.21 a2rs0	24
7.2.3.22 a2rs1	24
7.2.3.23 a2rs2	24
7.2.3.24 a2rs3	25
7.2.3.25 f	25
7.2.3.26 ls0	25
7.2.3.27 ls1	25
7.2.3.28 ls2	25

7.2.3.29 m	25
7.2.3.30 rs0	25
7.2.3.31 rs1	25
7.2.3.32 rs2	25
8 File Documentation	27
8.1 DNAStrand.py File Reference	27
8.2 DNAStrandTest.py File Reference	27
Index	29

Chapter 1

DNAStrand.py: AD1 Programação com Interfaces Gráficas 2020.1

DNAStrand This file implements the class **DNAStrand**, simulating parts of a DNA's bases sequence, with the main purpose of checking matches between two different strands.

Author

Júlia Mizarela (based on Professor Paulo Roma's work)

Since

16/03/2020

1.1 Introduction

DNAStrand.py: AD1 Programação com Interfaces Gráficas 2020.1 Professor: Paulo Roma Student: Júlia Mizarela de Oliveira Silva Student's ID #: 19113050516

1.2 Software

The following were used: PyCharm (IDE), TexWorks (LaTeX), unittest (testing), and Doxygen (documentation)

1.3 References

The following were consulted:

<https://docs.python.org/3/library/unittest.html>

<https://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>

<https://texfaq.org/FAQ-man-latex>

<https://pymotw.com/2/unittest/>

<http://opencdss.state.co.us/statemod/latest/doc-dev/project-init/doc-doxygen/>

<http://alesnosek.com/blog/2015/06/14/technical-documentation-with-doxygen/>

<http://www.doxygen.nl/manual/>

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

DNAStrand	11
DNAStrandTest	12

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

DNAStrand.DNAStrand	13
TestCase	
DNAStrandTest.DNAStrandClass	18

Chapter 4

Class Index

4.1 Class List

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

DNAStrand.DNAStrand	13
DNAStrandTest.DNAStrandClass	
DNAStrandClass unittest Class for testing all the methods of DNAStrand class	18

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

DNAStrand.py	27
DNAStrandTest.py	27

Chapter 6

Namespace Documentation

6.1 DNAStrand Namespace Reference

Classes

- class [DNAStrand](#)

Functions

- def [main](#) (args=None)
Main program for testing.

6.1.1 Function Documentation

6.1.1.1 main()

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

Main program for testing.

Parameters

<i>args</i>	two DNA strands.
-------------	------------------

6.2 DNAStrandTest Namespace Reference

Classes

- class [DNAStrandClass](#)
[DNAStrandClass](#) unittest Class for testing all the methods of [DNAStrand](#) class.

Functions

- def [main](#) (args=None)
Main program for testing.

6.2.1 Detailed Description

Class for testing the DNA strand matching.

AD1 Programação com Interfaces Gráficas 2020.1 Professor: Paulo Roma

Author

Júlia Mizarela (based on Professor Paulo Roma's work)

Since

16/03/2020

6.2.2 Function Documentation

6.2.2.1 main()

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

Main program for testing.

Parameters

<i>args</i>	two DNA strands.
-------------	------------------

Chapter 7

Class Documentation

7.1 DNASTrand.DNASTrand Class Reference

Public Member Functions

- def `__init__` (self, givenData)
DNASTrand class constructor.
- 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.

Static Public Attributes

- string `symbols` = 'ATCG'
Valid DNA symbols.

7.1.1 Constructor & Destructor Documentation

7.1.1.1 `__init__()`

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

`DNAStrand` class constructor.

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

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

Parameters

<code>givenData</code>	string of characters for this <code>DNAStrand</code> .
------------------------	--

7.1.2 Member Function Documentation

7.1.2.1 `__str__()`

```
def DNAStrand.DNAStrand.__str__ (
    self )
```

Returns a string representing the strand data of this `DNAStrand`.

7.1.2.2 `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

<i>other</i>	given DNAStrand to match with this strand.
<i>shift</i>	number of positions to shift other to the left.

Returns

number of matching pairs.

7.1.2.3 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

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

Returns

number of matching pairs.

7.1.2.4 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.

7.1.2.5 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.

Simulates the movement of a second DNA strand relative to a fixed one, comparing base (char) by base

Parameters

<i>other</i>	given DNAStrand , that's is going to be "shifting".
<i>shift</i>	number of positions to shift other strand to the left.

Returns

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

7.1.2.6 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.

Simulates the movement of a second DNA strand relative to a fixed one, comparing base (char) by base

Parameters

<i>other</i>	given DNAStrand .
<i>shift</i>	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.

7.1.2.7 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

<i>other</i>	given DNAStrand to be matched with this one.
--------------	--

Returns

maximum number of matching pairs.

7.1.2.8 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.

7.1.2.9 letterCount()

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

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

Parameters

<i>ch</i>	given character.
-----------	------------------

Returns

number of occurrences of *ch*.

7.1.2.10 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

<i>c1</i>	first character.
<i>c2</i>	second character.

Returns

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

7.1.3 Member Data Documentation

7.1.3.1 strand

```
DNAStrand.DNAStrand.strand
```

Strand of this DNA, in upper case.

7.1.3.2 symbols

```
string DNAStrand.DNAStrand.symbols = 'ATCG' [static]
```

Valid DNA symbols.

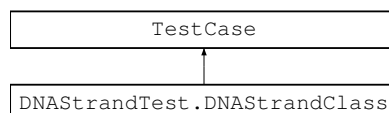
The documentation for this class was generated from the following file:

- [DNAStrand.py](#)

7.2 DNAStrandTest.DNAStrandClass Class Reference

[DNAStrandClass](#) Unittest Class for testing all the methods of [DNAStrand](#) class.

Inheritance diagram for DNAStrandTest.DNAStrandClass:



Public Member Functions

- def [setUp](#) (self)
Prepares four strings for the subsequent tests.
- def [test_isValid](#) (self)
Testing of isValid.
- def [test_createComplement](#) (self)
Testing of createComplement.
- def [test_countMatchesWithLeftShift](#) (self)
Testing of countMatchesWithLeftShift.
- def [test_findMatchesWithLeftShift](#) (self)
Testing of findMatchesWithLeftShift.
- def [test_countMatchesWithRightShift](#) (self)
Testing of countMatchesWithRightShift.
- def [test_findMatchesWithRightShift](#) (self)
Testing of findMatchesWithRightShift.
- def [test_findMaxPossibleMatches](#) (self)
Testing of findMaxPossibleMatches.
- def [test_letterCount](#) (self)
Testing of letterCount.
- def [test_matches](#) (self)
Testing of matches.

Public Attributes

- [f](#)
- [m](#)
- [a1](#)
- [a2](#)
- [ls0](#)
- [ls1](#)
- [ls2](#)
- [rs0](#)
- [rs1](#)
- [rs2](#)
- [a2ls0](#)
- [a2ls1](#)
- [a2ls2](#)
- [a2ls3](#)
- [a2ls4](#)
- [a2ls5](#)
- [a2ls6](#)
- [a2rs0](#)
- [a2rs1](#)
- [a2rs2](#)
- [a2rs3](#)
- [a1rs0](#)
- [a1rs1](#)
- [a1rs2](#)
- [a1rs3](#)
- [a1rs4](#)
- [a1rs5](#)
- [a1rs6](#)
- [a1ls0](#)
- [a1ls1](#)
- [a1ls2](#)
- [a1ls3](#)

7.2.1 Detailed Description

[DNAStrandClass](#) unittest Class for testing all the methods of [DNAStrand](#) class.

7.2.2 Member Function Documentation

7.2.2.1 setUp()

```
def DNAStrandTest.DNAStrandClass.setUp (
    self )
```

Prepares four strings for the subsequent tests.

7.2.2.2 test_countMatchesWithLeftShift()

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

Testing of countMatchesWithLeftShift.

7.2.2.3 test_countMatchesWithRightShift()

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

Testing of countMatchesWithRightShift.

7.2.2.4 test_createComplement()

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

Testing of createComplement.

7.2.2.5 test_findMatchesWithLeftShift()

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

Testing of findMatchesWithLeftShift.

7.2.2.6 test_findMatchesWithRightShift()

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

Testing of findMatchesWithRightShift.

7.2.2.7 test_findMaxPossibleMatches()

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

Testing of findMaxPossibleMatches.

7.2.2.8 test_isValid()

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

Testing of isValid.

7.2.2.9 test_letterCount()

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

Testing of letterCount.

7.2.2.10 test_matches()

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

Testing of matches.

7.2.3 Member Data Documentation

7.2.3.1 a1

`DNAStrandTest.DNAStrandClass.a1`

7.2.3.2 a1ls0

`DNAStrandTest.DNAStrandClass.a1ls0`

7.2.3.3 a1ls1

`DNAStrandTest.DNAStrandClass.a1ls1`

7.2.3.4 a1ls2

`DNAStrandTest.DNAStrandClass.a1ls2`

7.2.3.5 a1ls3

`DNAStrandTest.DNAStrandClass.a1ls3`

7.2.3.6 a1rs0

`DNAStrandTest.DNAStrandClass.a1rs0`

7.2.3.7 a1rs1

`DNAStrandTest.DNAStrandClass.a1rs1`

7.2.3.8 a1rs2

`DNAStrandTest.DNAStrandClass.a1rs2`

7.2.3.9 a1rs3

`DNAStrandTest.DNAStrandClass.a1rs3`

7.2.3.10 a1rs4

`DNAStrandTest.DNAStrandClass.a1rs4`

7.2.3.11 a1rs5

`DNAStrandTest.DNAStrandClass.a1rs5`

7.2.3.12 a1rs6

`DNAStrandTest.DNAStrandClass.a1rs6`

7.2.3.13 a2

`DNAStrandTest.DNAStrandClass.a2`

7.2.3.14 a2ls0

`DNAStrandTest.DNAStrandClass.a2ls0`

7.2.3.15 a2ls1

`DNAStrandTest.DNAStrandClass.a2ls1`

7.2.3.16 a2ls2

`DNAstrandTest.DNAstrandClass.a2ls2`

7.2.3.17 a2ls3

`DNAstrandTest.DNAstrandClass.a2ls3`

7.2.3.18 a2ls4

`DNAstrandTest.DNAstrandClass.a2ls4`

7.2.3.19 a2ls5

`DNAstrandTest.DNAstrandClass.a2ls5`

7.2.3.20 a2ls6

`DNAstrandTest.DNAstrandClass.a2ls6`

7.2.3.21 a2rs0

`DNAstrandTest.DNAstrandClass.a2rs0`

7.2.3.22 a2rs1

`DNAstrandTest.DNAstrandClass.a2rs1`

7.2.3.23 a2rs2

`DNAstrandTest.DNAstrandClass.a2rs2`

7.2.3.24 a2rs3

`DNAStrandTest.DNAStrandClass.a2rs3`

7.2.3.25 f

`DNAStrandTest.DNAStrandClass.f`

7.2.3.26 ls0

`DNAStrandTest.DNAStrandClass.ls0`

7.2.3.27 ls1

`DNAStrandTest.DNAStrandClass.ls1`

7.2.3.28 ls2

`DNAStrandTest.DNAStrandClass.ls2`

7.2.3.29 m

`DNAStrandTest.DNAStrandClass.m`

7.2.3.30 rs0

`DNAStrandTest.DNAStrandClass.rs0`

7.2.3.31 rs1

`DNAStrandTest.DNAStrandClass.rs1`

7.2.3.32 rs2

`DNAStrandTest.DNAStrandClass.rs2`

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

- [DNAStrandTest.py](#)

Chapter 8

File Documentation

8.1 DNASTrand.py File Reference

Classes

- class [DNASTrand.DNASTrand](#)

Namespaces

- [DNASTrand](#)

Functions

- def [DNASTrand.main](#) (args=None)
Main program for testing.

8.2 DNASTrandTest.py File Reference

Classes

- class [DNASTrandTest.DNASTrandClass](#)
[DNASTrandClass](#) unittest Class for testing all the methods of [DNASTrand](#) class.

Namespaces

- [DNASTrandTest](#)

Functions

- def [DNASTrandTest.main](#) (args=None)
Main program for testing.

Index

- `__init__`
 - `DNAStrand.DNAStrand`, [14](#)
 - `__str__`
 - `DNAStrand.DNAStrand`, [14](#)
- `a1`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1ls0`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1ls1`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1ls2`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1ls3`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1rs0`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1rs1`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1rs2`
 - `DNAStrandTest.DNAStrandClass`, [22](#)
- `a1rs3`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a1rs4`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a1rs5`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a1rs6`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a2`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a2ls0`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a2ls1`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a2ls2`
 - `DNAStrandTest.DNAStrandClass`, [23](#)
- `a2ls3`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2ls4`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2ls5`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2ls6`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2rs0`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2rs1`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2rs2`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `a2rs3`
 - `DNAStrandTest.DNAStrandClass`, [24](#)
- `countMatchesWithLeftShift`
 - `DNAStrand.DNAStrand`, [14](#)
- `countMatchesWithRightShift`
 - `DNAStrand.DNAStrand`, [15](#)
- `createComplement`
 - `DNAStrand.DNAStrand`, [15](#)
- `DNAStrand`, [11](#)
 - `main`, [11](#)
- `DNAStrand.DNAStrand`, [13](#)
 - `__init__`, [14](#)
 - `__str__`, [14](#)
 - `countMatchesWithLeftShift`, [14](#)
 - `countMatchesWithRightShift`, [15](#)
 - `createComplement`, [15](#)
 - `findMatchesWithLeftShift`, [15](#)
 - `findMatchesWithRightShift`, [16](#)
 - `findMaxPossibleMatches`, [16](#)
 - `isValid`, [17](#)
 - `letterCount`, [17](#)
 - `matches`, [17](#)
 - `strand`, [18](#)
 - `symbols`, [18](#)
- `DNAStrand.py`, [27](#)
- `DNAStrandTest`, [12](#)
 - `main`, [12](#)
- `DNAStrandTest.DNAStrandClass`, [18](#)
 - `a1`, [22](#)
 - `a1ls0`, [22](#)
 - `a1ls1`, [22](#)
 - `a1ls2`, [22](#)
 - `a1ls3`, [22](#)
 - `a1rs0`, [22](#)
 - `a1rs1`, [22](#)
 - `a1rs2`, [22](#)
 - `a1rs3`, [23](#)
 - `a1rs4`, [23](#)
 - `a1rs5`, [23](#)
 - `a1rs6`, [23](#)
 - `a2`, [23](#)
 - `a2ls0`, [23](#)
 - `a2ls1`, [23](#)
 - `a2ls2`, [23](#)
 - `a2ls3`, [24](#)
 - `a2ls4`, [24](#)
 - `a2ls5`, [24](#)

- a2ls6, [24](#)
- a2rs0, [24](#)
- a2rs1, [24](#)
- a2rs2, [24](#)
- a2rs3, [24](#)
- f, [25](#)
- ls0, [25](#)
- ls1, [25](#)
- ls2, [25](#)
- m, [25](#)
- rs0, [25](#)
- rs1, [25](#)
- rs2, [25](#)
- setUp, [20](#)
- test_countMatchesWithLeftShift, [20](#)
- test_countMatchesWithRightShift, [20](#)
- test_createComplement, [20](#)
- test_findMatchesWithLeftShift, [20](#)
- test_findMatchesWithRightShift, [21](#)
- test_findMaxPossibleMatches, [21](#)
- test_isValid, [21](#)
- test_letterCount, [21](#)
- test_matches, [21](#)
- DNAStrandTest.py, [27](#)
- f
 - DNAStrandTest.DNAStrandClass, [25](#)
- findMatchesWithLeftShift
 - DNAStrand.DNAStrand, [15](#)
- findMatchesWithRightShift
 - DNAStrand.DNAStrand, [16](#)
- findMaxPossibleMatches
 - DNAStrand.DNAStrand, [16](#)
- isValid
 - DNAStrand.DNAStrand, [17](#)
- letterCount
 - DNAStrand.DNAStrand, [17](#)
- ls0
 - DNAStrandTest.DNAStrandClass, [25](#)
- ls1
 - DNAStrandTest.DNAStrandClass, [25](#)
- ls2
 - DNAStrandTest.DNAStrandClass, [25](#)
- m
 - DNAStrandTest.DNAStrandClass, [25](#)
- main
 - DNAStrand, [11](#)
 - DNAStrandTest, [12](#)
- matches
 - DNAStrand.DNAStrand, [17](#)
- rs0
 - DNAStrandTest.DNAStrandClass, [25](#)
- rs1
 - DNAStrandTest.DNAStrandClass, [25](#)
- rs2
 - DNAStrandTest.DNAStrandClass, [25](#)
- setUp
 - DNAStrandTest.DNAStrandClass, [20](#)
- strand
 - DNAStrand.DNAStrand, [18](#)
- symbols
 - DNAStrand.DNAStrand, [18](#)
- test_countMatchesWithLeftShift
 - DNAStrandTest.DNAStrandClass, [20](#)
- test_countMatchesWithRightShift
 - DNAStrandTest.DNAStrandClass, [20](#)
- test_createComplement
 - DNAStrandTest.DNAStrandClass, [20](#)
- test_findMatchesWithLeftShift
 - DNAStrandTest.DNAStrandClass, [20](#)
- test_findMatchesWithRightShift
 - DNAStrandTest.DNAStrandClass, [21](#)
- test_findMaxPossibleMatches
 - DNAStrandTest.DNAStrandClass, [21](#)
- test_isValid
 - DNAStrandTest.DNAStrandClass, [21](#)
- test_letterCount
 - DNAStrandTest.DNAStrandClass, [21](#)
- test_matches
 - DNAStrandTest.DNAStrandClass, [21](#)