

DNAStrand

1.0

Generated by Doxygen 1.8.16

1 Namespace Index	2
1.1 Namespace List	2
2 Hierarchical Index	2
2.1 Class Hierarchy	2
3 Class Index	2
3.1 Class List	2
4 File Index	2
4.1 File List	2
5 Namespace Documentation	3
5.1 DNAStrand Namespace Reference	3
5.1.1 Detailed Description	3
5.1.2 Function Documentation	3
5.2 DNAStrandTest Namespace Reference	4
5.2.1 Detailed Description	4
5.2.2 Function Documentation	4
5.2.3 Variable Documentation	4
5.3 move Namespace Reference	5
5.3.1 Detailed Description	5
5.3.2 Function Documentation	5
6 Class Documentation	6
6.1 DNAStrand.DNAStrand Class Reference	6
6.1.1 Constructor & Destructor Documentation	7
6.1.2 Member Function Documentation	7
6.1.3 Member Data Documentation	11
6.2 DNAStrandTest.DNAStrandTest Class Reference	11
6.2.1 Detailed Description	12
6.2.2 Member Function Documentation	12
6.2.3 Member Data Documentation	14
6.3 move.GFG Class Reference	15
6.3.1 Detailed Description	17
6.3.2 Constructor & Destructor Documentation	17
6.3.3 Member Function Documentation	17
6.3.4 Member Data Documentation	20
7 File Documentation	24
7.1 DNAStrand.py File Reference	24
7.2 DNAStrandTest.py File Reference	25
7.3 move.py File Reference	25
Index	27

1 Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

DNAStrand	3
DNAStrandTest	4
move	5

2 Hierarchical Index

2.1 Class Hierarchy

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

DNAStrand.DNAStrand	6
move.GFG TestCase	15
DNAStrandTest.DNAStrandTest	11

3 Class Index

3.1 Class List

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

DNAStrand.DNAStrand	6
DNAStrandTest.DNAStrandTest Class for testing certain aspects of the behavior of DNAStrand	11
move.GFG Class for creating two DNA Strands and matching them, by moving the second one to the right or to the left	15

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

DNAStrand.py	24
------------------------------	----

DNAStrandTest.py	25
move.py	25

5 Namespace Documentation

5.1 DNAStrand Namespace Reference

Classes

- class [DNAStrand](#)

Functions

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

5.1.1 Detailed Description

Playing with string matching.

Author

Paulo Roma

Since

15/12/2019

See also

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

5.1.2 Function Documentation

5.1.2.1 `main()`

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

Main program for testing.

Parameters

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

5.2 DNAStrandTest Namespace Reference

Classes

- class [DNAStrandTest](#)
Class for testing certain aspects of the behavior of [DNAStrand](#).

Functions

- def [main](#) ()

Variables

- bool [runUnit](#) = True
Whether to simulate unittest or not.

5.2.1 Detailed Description

Class for testing the DNA strand matching.

Author

Paulo Roma

Since

18/12/2019

See also

<https://docs.python.org/2/library/unittest.html>

5.2.2 Function Documentation

5.2.2.1 main() `def DNAStrandTest.main ()`

5.2.3 Variable Documentation

5.2.3.1 runUnit `bool DNAStrandTest.runUnit = True`

Whether to simulate unittest or not.

5.3 move Namespace Reference

Classes

- class [GFG](#)

Class for creating two DNA Strands and matching them, by moving the second one to the right or to the left.

Functions

- def [main](#) (argv=None)

Instantiates an object of class Tk, responsible for creating a tkinter toplevel window.

5.3.1 Detailed Description

Move and animate two [DNAStrand](#).

Author

Paulo Roma

Since

25/12/2019

See also

<https://www.geeksforgeeks.org/python-tkinter-moving-objects-using-canvas-move-method/>

5.3.2 Function Documentation

5.3.2.1 main()

```
def move.main (
    argv = None )
```

Instantiates an object of class Tk, responsible for creating a tkinter toplevel window.

Accepts, in the command line, four arguments:

Parameters

<i>argv</i>	command line arguments
	<ul style="list-style-type: none">• h help• n DNA1 size• m DNA2 size• v verbose mode

Usage:

- `move.py -n 6 -m 7 -v` or
- `move.py --dna1=6 --dna2=7 -v` or
- `move.py --help`

6 Class Documentation

6.1 DNASTrand.DNASTrand Class Reference

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.

Static Public Attributes

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

6.1.1 Constructor & Destructor Documentation

6.1.1.1 `__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

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

6.1.2 Member Function Documentation

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

Returns a string representing the strand data of this [DNAStrand](#).

References DNAStrand.DNAStrand.strand.

6.1.2.2 `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.

References DNAStrand.DNAStrand.findMatchesWithLeftShift().

Referenced by DNAStrand.DNAStrand.findMaxPossibleMatches().

6.1.2.3 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.

References `DNAStrand.DNAStrand.findMatchesWithRightShift()`.

Referenced by `DNAStrand.DNAStrand.findMaxPossibleMatches()`.

6.1.2.4 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.

References `DNAStrand.DNAStrand.strand`.

6.1.2.5 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.

Parameters

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

References DNAStrand.DNAStrand.matches(), and DNAStrand.DNAStrand.strand.

Referenced by DNAStrand.DNAStrand.countMatchesWithLeftShift().

6.1.2.6 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.

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.

References DNAStrand.DNAStrand.matches(), and DNAStrand.DNAStrand.strand.

Referenced by DNAStrand.DNAStrand.countMatchesWithRightShift().

6.1.2.7 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 and its position.

References DNAStrand.DNAStrand.countMatchesWithLeftShift(), DNAStrand.DNAStrand.countMatchesWith↔RightShift(), and DNAStrand.DNAStrand.strand.

6.1.2.8 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.

References DNASTrand.DNASTrand.strand.

6.1.2.9 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.

References DNASTrand.DNASTrand.strand.

6.1.2.10 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.

Referenced by DNASTrand.DNASTrand.findMatchesWithLeftShift(), and DNASTrand.DNASTrand.findMatchesWith↔RightShift().

6.1.3 Member Data Documentation

6.1.3.1 strand `DNAStrand.DNAStrand.strand`

Strand of this DNA, in upper case.

Referenced by `DNAStrand.DNAStrand.__str__()`, `DNAStrand.DNAStrand.createComplement()`, `DNAStrand.DNAStrand.findMatchesWithLeftShift()`, `DNAStrand.DNAStrand.findMatchesWithRightShift()`, `DNAStrand.DNAStrand.findMaxPossibleMatches()`, `DNAStrand.DNAStrand.isValid()`, and `DNAStrand.DNAStrand.letterCount()`.

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

Valid DNA symbols.

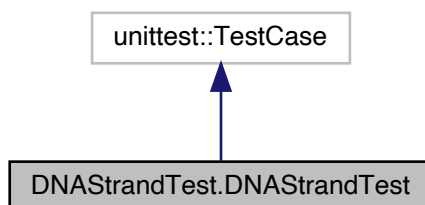
The documentation for this class was generated from the following file:

- [DNAStrand.py](#)

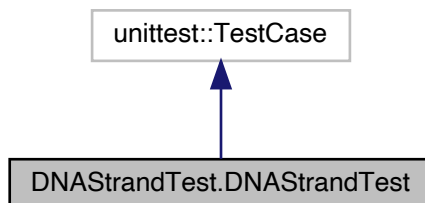
6.2 DNAStrandTest.DNAStrandTest Class Reference

Class for testing certain aspects of the behavior of [DNAStrand](#).

Inheritance diagram for `DNAStrandTest.DNAStrandTest`:



Collaboration diagram for `DNAStrandTest.DNAStrandTest`:



Public Member Functions

- def [assertEquals](#) (self, msg, expected_value, actual_value)
Asserts that two objects are equal.
- def [test_createComplement](#) (self)
- def [test_findMatchesWithRightShift](#) (self)
- def [test_countMatchesWithRightShift](#) (self)
- def [test_findMatchesWithLeftShift](#) (self)
- def [test_countMatchesWithLeftShift](#) (self)
- def [test_findMaxPossibleMatches](#) (self)
- def [test_letterCount](#) (self)
- def [test_isValid](#) (self)
- def [test_matches](#) (self)
- def [test_Exception](#) (self)

Static Public Attributes

- [d1](#) = [DNAStrand](#)("TCAT")
Two [DNAStrand](#) objects to be used in the tests.
- [d2](#) = [DNAStrand](#)("AGAGCAT")

6.2.1 Detailed Description

Class for testing certain aspects of the behavior of [DNAStrand](#).

6.2.2 Member Function Documentation

6.2.2.1 [assertEquals\(\)](#) `def DNAStrandTest.DNAStrandTest.assertEquals (`
self,
msg,
expected_value,
actual_value)

Asserts that two objects are equal.

If they are not, an `AssertionError` is thrown with the given message. If expected and actual are null, they are considered equal. This is borrowed from the JUnit API. Only failed assertions are recorded.

Parameters

<i>msg</i>	a string to be displayed in case of fail.
<i>expected_value</i>	what we expected to get.
<i>actual_value</i>	what the method actually gave us.

See also

<http://junit.sourceforge.net/javadoc/org/junit/Assert.html>

Referenced by DNAStrandTest.DNAStrandTest.test_countMatchesWithLeftShift(), DNAStrandTest.DNAStrandTest.test_countMatchesWithRightShift(), DNAStrandTest.DNAStrandTest.test_createComplement(), DNAStrandTest.DNAStrandTest.test_findMatchesWithLeftShift(), DNAStrandTest.DNAStrandTest.test_findMatchesWithRightShift(), DNAStrandTest.DNAStrandTest.test_findMaxPossibleMatches(), DNAStrandTest.DNAStrandTest.test_isValid(), DNAStrandTest.DNAStrandTest.test_letterCount(), and DNAStrandTest.DNAStrandTest.test_matches().

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

6.2.2.5 test_Exception() `def DNAStrandTest.DNAStrandTest.test_Exception (`
`self)`

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), and DNAStrandTest.DNAStrandTest.d1.

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

References DNAStrandTest.DNAStrandTest.assertEquals(), DNAStrandTest.DNAStrandTest.d1, and DNAStrandTest.DNAStrandTest.d2.

6.2.3 Member Data Documentation

6.2.3.1 d1 `DNAStrandTest.DNAStrandTest.d1 = DNAStrand("TCAT") [static]`

Two [DNAStrand](#) objects to be used in the tests.

Referenced by DNAStrandTest.DNAStrandTest.test_countMatchesWithLeftShift(), DNAStrandTest.DNAStrandTest.test_countMatchesWithRightShift(), DNAStrandTest.DNAStrandTest.test_createComplement(), DNAStrandTest.DNAStrandTest.test_findMatchesWithLeftShift(), DNAStrandTest.DNAStrandTest.test_findMatchesWithRightShift(), DNAStrandTest.DNAStrandTest.test_findMaxPossibleMatches(), DNAStrandTest.DNAStrandTest.test_isValid(), DNAStrandTest.DNAStrandTest.test_letterCount(), and DNAStrandTest.DNAStrandTest.test_matches().

6.2.3.2 d2 `DNAstrandTest.DNAstrandTest.d2 = DNAstrand("AGAGCAT") [static]`

Referenced by `DNAstrandTest.DNAstrandTest.test_countMatchesWithLeftShift()`, `DNAstrandTest.DNAstrandTest.test_countMatchesWithRightShift()`, `DNAstrandTest.DNAstrandTest.test_createComplement()`, `DNAstrandTest.DNAstrandTest.test_findMatchesWithLeftShift()`, `DNAstrandTest.DNAstrandTest.test_findMatchesWithRightShift()`, `DNAstrandTest.DNAstrandTest.test_findMaxPossibleMatches()`, `DNAstrandTest.DNAstrandTest.test_isValid()`, and `DNAstrandTest.DNAstrandTest.test_matches()`.

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

- [DNAstrandTest.py](#)

6.3 move.GFG Class Reference

Class for creating two DNA Strands and matching them, by moving the second one to the right or to the left.

Public Member Functions

- `def __init__ (self, master=None, n1=9, n2=9, debug=False)`
Constructor.
- `def movement (self)`
This is the move() method.
- `def left (self, event)`
For motion in negative x direction.
- `def right (self, event)`
For motion in positive x direction.
- `def up (self, event)`
For motion in positive y direction.
- `def down (self, event)`
For motion in negative y direction.
- `def exit (self, event)`
Finishes this program.
- `def reset (self, event)`
Resets the two DNAs to their initial positions.
- `def shuffle (self, event)`
Shuffles the second DNA.
- `def help (self, event)`
Help.
- `def resize (self, event)`
Resize the window.
- `def maximum (self, event)`
Go to the position of the maximum number of matches.

Public Attributes

- **master**
master window.
- **delta**
displacement in the horizontal direction.
- **deltaY**
displacement in the vertical direction.
- **posx**
text x position into the canvas for both DNAs.
- **posy**
text y position into the canvas for DNA1.
- **posy2**
text y position into the canvas for DNA2.
- **cx**
canvas width.
- **cy**
canvas height.
- **c**
canvas.
- **x**
to take care of movement in x direction.
- **y**
to take care of movement in y direction.
- **dx**
initial horizontal position in characters.
- **dy**
initial vertical position in characters.
- **debug**
debugging mode.
- **canvas**
canvas object to create graphical shapes.
- **font**
creating text using a mono spaced font (constant width).
- **t1**
text for DNA1.
- **t2**
text for DNA2.
- **text**
text object into the canvas.
- **text2**
text2 object into the canvas.
- **textBox**
text box object into the canvas.
- **text2Box**
text2 box object into the canvas.
- **tt1**
text for DNA1 with extra spaces between letters.
- **tt2**
text for DNA2 with extra spaces between letters.
- **deltaU**

- `deltaL`
hold the size of an upper letter.
- `dna1`
hold the size of a lower letter.
- `dna2`
DNA1.
- `dna2`
DNA1.

6.3.1 Detailed Description

Class for creating two DNA Strands and matching them, by moving the second one to the right or to the left.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 `__init__()` `def move.GFG.__init__ (`
`self,`
`master = None,`
`n1 = 9,`
`n2 = 9,`
`debug = False)`

Constructor.

Creates two DNA strands of the given size, using random symbols in each position. If a size is less than or equal 0, then a predefined string is used instead.

Parameters

<i>master</i>	root window.
<i>n1</i>	size of DNA1.
<i>n2</i>	size of DNA2.
<i>debug</i>	toggle debugging mode.

6.3.3 Member Function Documentation

6.3.3.1 `down()` `def move.GFG.down (`
`self,`
`event)`

For motion in negative y direction.

References `move.GFG.debug`, `move.GFG.deltaY`, `move.GFG.movement()`, `move.GFG.x`, and `move.GFG.y`.

6.3.3.2 exit() `def move.GFG.exit (`
 `self,`
 `event)`

Finishes this program.

6.3.3.3 help() `def move.GFG.help (`
 `self,`
 `event)`

Help.

6.3.3.4 left() `def move.GFG.left (`
 `self,`
 `event)`

For motion in negative x direction.

References `move.GFG.debug`, `move.GFG.delta`, `move.GFG.movement()`, `move.GFG.x`, and `move.GFG.y`.

Referenced by `move.GFG.maximum()`.

6.3.3.5 maximum() `def move.GFG.maximum (`
 `self,`
 `event)`

Go to the position of the maximum number of matches.

References `move.GFG.dna1`, `move.GFG.dna2`, `move.GFG.left()`, `move.GFG.reset()`, and `move.GFG.right()`.

6.3.3.6 movement() `def move.GFG.movement (`
 `self)`

This is the `move()` method.

Translates the second DNA in four directions: left, right, up or down.

References `move.GFG.canvas`, `move.GFG.cx`, `move.GFG.cy`, `move.GFG.debug`, `move.GFG.dna1`, `move.GFG.dna2`, `move.GFG.dx`, `move.GFG.dy`, `move.GFG.master`, `move.GFG.reset()`, `move.GFG.text`, `move.GFG.text2`, `move.GFG.text2Box`, `move.GFG.textBox`, `move.GFG.x`, and `move.GFG.y`.

Referenced by `move.GFG.down()`, `move.GFG.left()`, `move.GFG.reset()`, `move.GFG.right()`, and `move.GFG.up()`.

6.3.3.7 reset()

```
def move.GFG.reset (
    self,
    event )
```

Resets the two DNAs to their initial positions.

References move.GFG.canvas, move.GFG.debug, move.GFG.dx, move.GFG.dy, move.GFG.font, move.GFG.movement(), move.GFG.posx, move.GFG.posy, move.GFG.posy2, move.GFG.text, move.GFG.text2, move.GFG.text2Box, move.GFG.textBox, move.GFG.tt1, move.GFG.tt2, move.GFG.x, and move.GFG.y.

Referenced by move.GFG.maximum(), move.GFG.movement(), move.GFG.resize(), and move.GFG.shuffle().

6.3.3.8 resize()

```
def move.GFG.resize (
    self,
    event )
```

Resize the window.

References move.GFG.canvas, move.GFG.cx, move.GFG.cy, move.GFG.deltaY, move.GFG.posx, move.GFG.posy, move.GFG.posy2, move.GFG.reset(), and move.GFG.text.

6.3.3.9 right()

```
def move.GFG.right (
    self,
    event )
```

For motion in positive x direction.

References move.GFG.debug, move.GFG.delta, move.GFG.movement(), move.GFG.x, and move.GFG.y.

Referenced by move.GFG.maximum().

6.3.3.10 shuffle()

```
def move.GFG.shuffle (
    self,
    event )
```

Shuffles the second DNA.

References move.GFG.debug, move.GFG.dna2, move.GFG.reset(), move.GFG.tt2, and move.GFG.tt2.

6.3.3.11 up()

```
def move.GFG.up (
    self,
    event )
```

For motion in positive y direction.

References move.GFG.debug, move.GFG.deltaY, move.GFG.movement(), move.GFG.x, and move.GFG.y.

6.3.4 Member Data Documentation

6.3.4.1 **c** `move.GFG.c`

canvas.

6.3.4.2 **canvas** `move.GFG.canvas`

canvas object to create graphical shapes.

Referenced by `move.GFG.movement()`, `move.GFG.reset()`, and `move.GFG.resize()`.

6.3.4.3 **cx** `move.GFG.cx`

canvas width.

Referenced by `move.GFG.movement()`, and `move.GFG.resize()`.

6.3.4.4 **cy** `move.GFG.cy`

canvas height.

Referenced by `move.GFG.movement()`, and `move.GFG.resize()`.

6.3.4.5 **debug** `move.GFG.debug`

debugging mode.

Referenced by `move.GFG.down()`, `move.GFG.left()`, `move.GFG.movement()`, `move.GFG.reset()`, `move.GFG.right()`, `move.GFG.shuffle()`, and `move.GFG.up()`.

6.3.4.6 **delta** `move.GFG.delta`

displacement in the horizontal direction.

should be the width of a single character.

Referenced by `move.GFG.left()`, and `move.GFG.right()`.

6.3.4.7 deltaL `move.GFG.deltaL`

hold the size of a lower letter.

6.3.4.8 deltaU `move.GFG.deltaU`

hold the size of an upper letter.

6.3.4.9 deltaY `move.GFG.deltaY`

displacement in the vertical direction.

should be the height of a single character.

Referenced by `move.GFG.down()`, `move.GFG.resize()`, and `move.GFG.up()`.

6.3.4.10 dna1 `move.GFG.dna1`

DNA1.

Referenced by `move.GFG.maximum()`, and `move.GFG.movement()`.

6.3.4.11 dna2 `move.GFG.dna2`

DNA1.

Referenced by `move.GFG.maximum()`, `move.GFG.movement()`, and `move.GFG.shuffle()`.

6.3.4.12 dx `move.GFG.dx`

initial horizontal position in characters.

Referenced by `move.GFG.movement()`, and `move.GFG.reset()`.

6.3.4.13 dy `move.GFG.dy`

initial vertical position in characters.

Referenced by `move.GFG.movement()`, and `move.GFG.reset()`.

6.3.4.14 font `move.GFG.font`

creating text using a mono spaced font (constant width).

```
self.font = tkFont.Font(font='TkFixedFont',size=80,weight="bold")
```

Referenced by `move.GFG.reset()`.

6.3.4.15 master `move.GFG.master`

master window.

Referenced by `move.GFG.movement()`.

6.3.4.16 posx `move.GFG.posx`

text x position into the canvas for both DNAs.

Referenced by `move.GFG.reset()`, and `move.GFG.resize()`.

6.3.4.17 posy `move.GFG.posy`

text y position into the canvas for DNA1.

Referenced by `move.GFG.reset()`, and `move.GFG.resize()`.

6.3.4.18 posy2 `move.GFG.posy2`

text y position into the canvas for DNA2.

Referenced by `move.GFG.reset()`, and `move.GFG.resize()`.

6.3.4.19 t1 `move.GFG.t1`

text for DNA1.

6.3.4.20 t2 `move.GFG.t2`

text for DNA2.

Referenced by `move.GFG.shuffle()`.

6.3.4.21 text `move.GFG.text`

text object into the canvas.

Referenced by `move.GFG.movement()`, `move.GFG.reset()`, and `move.GFG.resize()`.

6.3.4.22 text2 `move.GFG.text2`

text2 object into the canvas.

Referenced by `move.GFG.movement()`, and `move.GFG.reset()`.

6.3.4.23 text2Box `move.GFG.text2Box`

text2 box object into the canvas.

Referenced by `move.GFG.movement()`, and `move.GFG.reset()`.

6.3.4.24 textBox `move.GFG.textBox`

text box object into the canvas.

Referenced by `move.GFG.movement()`, and `move.GFG.reset()`.

6.3.4.25 tt1 `move.GFG.tt1`

text for DNA1 with extra spaces between letters.

Referenced by `move.GFG.reset()`.

6.3.4.26 `tt2` `move.GFG.tt2`

text for DNA2 with extra spaces between letters.

Referenced by `move.GFG.reset()`, and `move.GFG.shuffle()`.

6.3.4.27 `x` `move.GFG.x`

to take care of movement in x direction.

Referenced by `move.GFG.down()`, `move.GFG.left()`, `move.GFG.movement()`, `move.GFG.reset()`, `move.GFG.right()`, and `move.GFG.up()`.

6.3.4.28 `y` `move.GFG.y`

to take care of movement in y direction.

Referenced by `move.GFG.down()`, `move.GFG.left()`, `move.GFG.movement()`, `move.GFG.reset()`, `move.GFG.right()`, and `move.GFG.up()`.

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

- [move.py](#)

7 File Documentation

7.1 DNAStrand.py File Reference

Classes

- class [DNAStrand.DNAStrand](#)

Namespaces

- [DNAStrand](#)

Functions

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

7.2 DNAStrandTest.py File Reference

Classes

- class [DNAStrandTest.DNAStrandTest](#)
Class for testing certain aspects of the behavior of [DNAStrand](#).

Namespaces

- [DNAStrandTest](#)

Functions

- def [DNAStrandTest.main](#) ()

Variables

- bool [DNAStrandTest.runUnit](#) = True
Whether to simulate unittest or not.

7.3 move.py File Reference

Classes

- class [move.GFG](#)
Class for creating two DNA Strands and matching them, by moving the second one to the right or to the left.

Namespaces

- [move](#)

Functions

- def [move.main](#) (argv=None)
Instantiates an object of class Tk, responsible for creating a tkinter toplevel window.

Index

- `__init__`
 - `DNAStrand.DNAStrand`, 7
 - `move.GFG`, 17
 - `__str__`
 - `DNAStrand.DNAStrand`, 7
- `assertEquals`
 - `DNAStrandTest.DNAStrandTest`, 12
- `c`
 - `move.GFG`, 20
- `canvas`
 - `move.GFG`, 20
- `countMatchesWithLeftShift`
 - `DNAStrand.DNAStrand`, 7
- `countMatchesWithRightShift`
 - `DNAStrand.DNAStrand`, 7
- `createComplement`
 - `DNAStrand.DNAStrand`, 8
- `cx`
 - `move.GFG`, 20
- `cy`
 - `move.GFG`, 20
- `d1`
 - `DNAStrandTest.DNAStrandTest`, 14
- `d2`
 - `DNAStrandTest.DNAStrandTest`, 14
- `debug`
 - `move.GFG`, 20
- `delta`
 - `move.GFG`, 20
- `deltaL`
 - `move.GFG`, 20
- `deltaU`
 - `move.GFG`, 21
- `deltaY`
 - `move.GFG`, 21
- `dna1`
 - `move.GFG`, 21
- `dna2`
 - `move.GFG`, 21
- `DNAStrand`, 3
 - `main`, 3
- `DNAStrand.DNAStrand`, 6
 - `__init__`, 7
 - `__str__`, 7
 - `countMatchesWithLeftShift`, 7
 - `countMatchesWithRightShift`, 7
 - `createComplement`, 8
 - `findMatchesWithLeftShift`, 8
 - `findMatchesWithRightShift`, 9
 - `findMaxPossibleMatches`, 9
 - `isValid`, 9
 - `letterCount`, 10
 - `matches`, 10
 - `strand`, 11
 - `symbols`, 11
- `DNAStrand.py`, 24
- `DNAStrandTest`, 4
 - `main`, 4
 - `runUnit`, 4
- `DNAStrandTest.DNAStrandTest`, 11
 - `assertEquals`, 12
 - `d1`, 14
 - `d2`, 14
 - `test_countMatchesWithLeftShift`, 13
 - `test_countMatchesWithRightShift`, 13
 - `test_createComplement`, 13
 - `test_Exception`, 13
 - `test_findMatchesWithLeftShift`, 13
 - `test_findMatchesWithRightShift`, 13
 - `test_findMaxPossibleMatches`, 14
 - `test_isValid`, 14
 - `test_letterCount`, 14
 - `test_matches`, 14
- `DNAStrandTest.py`, 25
- `down`
 - `move.GFG`, 17
- `dx`
 - `move.GFG`, 21
- `dy`
 - `move.GFG`, 21
- `exit`
 - `move.GFG`, 17
- `findMatchesWithLeftShift`
 - `DNAStrand.DNAStrand`, 8
- `findMatchesWithRightShift`
 - `DNAStrand.DNAStrand`, 9
- `findMaxPossibleMatches`
 - `DNAStrand.DNAStrand`, 9
- `font`
 - `move.GFG`, 21
- `help`
 - `move.GFG`, 18
- `isValid`
 - `DNAStrand.DNAStrand`, 9
- `left`
 - `move.GFG`, 18
- `letterCount`
 - `DNAStrand.DNAStrand`, 10
- `main`
 - `DNAStrand`, 3
 - `DNAStrandTest`, 4
 - `move`, 5
- `master`
 - `move.GFG`, 22

- matches
 - DNAStrand.DNAStrand, 10
- maximum
 - move.GFG, 18
- move, 5
 - main, 5
- move.GFG, 15
 - __init__, 17
 - c, 20
 - canvas, 20
 - cx, 20
 - cy, 20
 - debug, 20
 - delta, 20
 - deltaL, 20
 - deltaU, 21
 - deltaY, 21
 - dna1, 21
 - dna2, 21
 - down, 17
 - dx, 21
 - dy, 21
 - exit, 17
 - font, 21
 - help, 18
 - left, 18
 - master, 22
 - maximum, 18
 - movement, 18
 - posx, 22
 - posy, 22
 - posy2, 22
 - reset, 18
 - resize, 19
 - right, 19
 - shuffle, 19
 - t1, 22
 - t2, 22
 - text, 23
 - text2, 23
 - text2Box, 23
 - textBox, 23
 - tt1, 23
 - tt2, 23
 - up, 19
 - x, 24
 - y, 24
- move.py, 25
- movement
 - move.GFG, 18
- posx
 - move.GFG, 22
- posy
 - move.GFG, 22
- posy2
 - move.GFG, 22
- reset
 - move.GFG, 18
- resize
 - move.GFG, 19
- right
 - move.GFG, 19
- runUnit
 - DNAStrandTest, 4
- shuffle
 - move.GFG, 19
- strand
 - DNAStrand.DNAStrand, 11
- symbols
 - DNAStrand.DNAStrand, 11
- t1
 - move.GFG, 22
- t2
 - move.GFG, 22
- test_countMatchesWithLeftShift
 - DNAStrandTest.DNAStrandTest, 13
- test_countMatchesWithRightShift
 - DNAStrandTest.DNAStrandTest, 13
- test_createComplement
 - DNAStrandTest.DNAStrandTest, 13
- test_Exception
 - DNAStrandTest.DNAStrandTest, 13
- test_findMatchesWithLeftShift
 - DNAStrandTest.DNAStrandTest, 13
- test_findMatchesWithRightShift
 - DNAStrandTest.DNAStrandTest, 13
- test_findMaxPossibleMatches
 - DNAStrandTest.DNAStrandTest, 14
- test_isValid
 - DNAStrandTest.DNAStrandTest, 14
- test_letterCount
 - DNAStrandTest.DNAStrandTest, 14
- test_matches
 - DNAStrandTest.DNAStrandTest, 14
- text
 - move.GFG, 23
- text2
 - move.GFG, 23
- text2Box
 - move.GFG, 23
- textBox
 - move.GFG, 23
- tt1
 - move.GFG, 23
- tt2
 - move.GFG, 23
- up
 - move.GFG, 19
- x
 - move.GFG, 24
- y
 - move.GFG, 24