Nucleotide String Library

1.0.0

Generated by Doxygen 1.8.6

Mon Dec 7 2015 18:33:26

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 Helix Class Reference

Public Member Functions

- Helix (void)

  No param constructor.
- Helix (string)

  string param constructor.
- Helix (NucleoString< DNA >)

  NucleoString<DNA> param constructor.
- Helix (NucleoString< DNA >, NucleoString< DNA >)

  NucleoString<DNA> dual param constructor.
- void fix (WHICH)
- NucleoString< DNA > getNucleoString (WHICH)

2.1.1 Constructor & Destructor Documentation

2.1.1.1 Helix::Helix (void)

No param constructor.

Creates 2 new NucleoString<DNA> objects, with no input string.

2.1.1.2 Helix::Helix (string *entry*)

string param constructor.

Creates a new NucleoString<DNA> from a DNA character string. It then creates its complement, by using the funcition and stores both in a Helxi object.

2.1.1.3 Helix::Helix (NucleoString< DNA > *entry*)

NucleoString<DNA> param constructor.

Creates a new NucleoString<DNA> by getting the complement using the NucleoString<DNA>::complement() function and stores both in a Helix object.

2.1.1.4 Helix::Helix (NucleoString< DNA > , NucleoString< DNA >)

NucleoString<DNA> dual param constructor.

Creates a Helix object by setting both of the NucleoString<DNA> chains. Warning: You may end up creating a invalid Helix as this constructor won't check if they are compatible chains.

2.1.2 Member Function Documentation

2.1.2.1 void Helix::fix (WHICH)

Fixes the current Helix by using one of the NucleoStrings<DNA> as a base. you may use the macros FIRST or SECOND to decide which one to fix. (It will use the other one as base)

2.1.2.2 NucleoString<DNA> Helix::getNucleoString (WHICH)

Returns the required NucleoString<DNA> as a new NucleoString<DNA> object. you may use the macros FIRST or SECOND to decide which on you want to return.

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

- /home/david/Documents/Estructuras/Nucleolib/juanjo/Nucleotide_String_Library/include/Helix.h
- /home/david/Documents/Estructuras/Nucleolib/juanjo/Nucleotide_String_Library/source/Helix.cpp

2.2 NucleoString< ANN > Class Template Reference

NucleoString Class.

`#include <NucleoString.h>`

Public Member Functions

- NucleoString (void)

  No param constructor.
- NucleoString (string)

  string param constructor
- NucleoString (const NucleoString< ANN > &)

  copy constructor
- NucleoString (NucleoString< RNA >)

  NucleoString<RNA> param constructor.
- NucleoString (NucleoString< DNA >)

  NucleoString<DNA> param constructor.
- string chainret (void)

  Return the Nucleotid String as an string of nucleobases for the current NucleoString.
- NBASE operator[] (void)

  Return the character value of the i nucleobase of the string.
- NucleoString< ANN > complement (void)

  Finds and creates a new NucleoString based on the complement of this NucleoString.
- NucleoString<!ANN > transcript (void)

  Finds and creates a new NucleoString based on the transcript form of this NucleoString.
- string NucleoString< ANN > getfasta (void)

  Transcripts a sequences of nucleotides into a RNA if it is a DNA or DNA if it is a ARN.

- NucleoString< ANN > headerret (void)

  Returns the header of the equivalent FASTA format String.
- NucleoString< ANN > cut (int n1, int n2)

  Cut a NucleoString from the point a to the point b.
- NucleoString< ANN > operator+ (NucleoString< ANN >)

  Concatenates 2 NucleoStrings together and create a new NucleoString containing both.
- int compare (NucleoString nucleo1, NucleoString nucleo2)

  Compare how similar are two nucleotide sequences one from other. It returns the percentage of similitude.
- template<>
  NucleoString (string entry)
- template<>
  NucleoString (string entry)
- template<>
  NucleoString (NucleoString< ARN > other)
- template<>
  NucleoString (NucleoString< ADN > other)
- template<>
  NucleoString< ARN > **transcript** ()

2.2.1 Detailed Description

template<NTYPE ANN>class NucleoString< ANN >

NucleoString Class.

This class implements a single DNA or RNA chain.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 template<NTYPE ANN> NucleoString< ANN >::NucleoString (void)

No param constructor.

/ Sets chain as a string with no characters.

2.2.2.2 template<NTYPE ANN> NucleoString< ANN >::NucleoString (string)

string param constructor

Sets the input string to the chain. Program will halt executing an exception if the strings values are not valid for the Nuclotid Chain (DNA or RNA).

2.2.2.3 template<NTYPE ANN> NucleoString< ANN >::NucleoString (const NucleoString< ANN > & *other*)

copy constructor

Creates a equal Nucleotid String from a already existing one.

2.2.2.4 template<NTYPE ANN> NucleoString< ANN >::NucleoString (NucleoString< RNA >)

NucleoString<RNA> param constructor.

Acts like a copy constructor if the type matches(Both of them are RNA chains). If not it will create a new DNA chain by finding the RNA's transcription. Warning: You may end up creating a DNA chain from a RNA chain if you're not careful enough.

2.2.2.5 template<NTYPE ANN> NucleoString< ANN >::NucleoString (NucleoString< DNA >)

NucleoString<DNA> param constructor.

Acts like a copy constructor if the type matches(Both of them are DNA chains). If not it will create a new RNA chain by finding the DNA's transcription. Warning: You may end up creating a DNA chain from a RNA chain if you're not careful enough.

2.2.3 Member Function Documentation

2.2.3.1 template<NTYPE ANN> NucleoString<ANN> NucleoString< ANN >::complement (void)

Finds and creates a new NucleoString based on the complement of this NucleoString.

This only creates the complement of the same type of NucleoString. If you want to create the complement in the transcription way (RNA -> DNA or DNA–> RNA) you should use the transcript function

2.2.3.2 template<NTYPE ANN> string NucleoString< ANN >::getfasta (void)

Transcripts a sequences of nucleotides into a RNA if it is a DNA or DNA if it is a ARN.

Returns the value of the FASTA format Nucleotid String

2.2.3.3 template<NTYPE ANN> NucleoString<!ANN> NucleoString< ANN >::transcript (void)

Finds and creates a new NucleoString based on the transcript form of this NucleoString.

This creates the complement of the diferent type of NucleoString. if you want to create the complement in the standard way (DNA-> DNA or RNA-> RNA) you should use the complement funciton

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

- /home/david/Documents/Estructuras/Nucleolib/juanjo/Nucleotide_String_Library/include/NucleoString.h
- /home/david/Documents/Estructuras/Nucleolib/juanjo/Nucleotide_String_Library/source/NucleoString.cpp

Index