

## BLAST Simple Example

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	blastobject Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	5
3.1.1.1	blastobject() . . . . .	5
3.1.2	Member Function Documentation . . . . .	6
3.1.2.1	comparador() . . . . .	6
3.1.2.2	gap() . . . . .	6
3.1.2.3	gapclear() . . . . .	7
3.1.2.4	raw() . . . . .	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	main.cpp File Reference . . . . .	9
4.1.1	Detailed Description . . . . .	10
	<b>Index</b>	<b>11</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">blastobject</a> . . . . .	5
---------------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

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

<b>blastobject.h</b>	??
<a href="#">main.cpp</a>	
En este programa se utiliza el algoritmo BLAST para comparacion de secuencias en su version mas simple, comparando con una base de datos creada por el usuario con el proposito de observar su comportamiento de algoritmo y sus resultados. En este caso dichos objetos son del tipo blastobject	9





## Chapter 3

# Class Documentation

### 3.1 blastobject Class Reference

#### Public Member Functions

- `blastobject ()`  
*Constructor vacio de clase blastobject.*
- `blastobject (const blastobject &orig)`  
*Constructor de la clase blastobject.*
- `virtual ~blastobject ()`  
*Destructor de la clase blastobject.*
- `char comparador (char **lista, int contador)`  
*Compara las secuencias de caracteres con la tabla de traduccion.*
- `int gapclear (char **lista)`  
*Retorna una secuencia con solo los caracteres similares.*
- `int raw (char **lista)`  
*Compara las secuencias de caracteres con la referencia y cuenta su puntuacion cruda.*
- `int gap (char **lista, int n)`  
*Compara las secuencias de caracteres con la referencia y cuenta sus vacios.*

#### Public Attributes

- `int rawscore`
- `int gapcounter`
- `int size`
- `const char * base`

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 blastobject()

```
blastobject::blastobject (  
    const blastobject & orig )
```

Constructor de la clase blastobject.

**Parameters**

<i>blastobject</i> &	Constante objeto.
----------------------	-------------------

**3.1.2 Member Function Documentation****3.1.2.1 comparador()**

```
char blastobject::comparador (  
    char ** lista,  
    int contador )
```

Compara las secuencias de caracteres con la tabla de traduccion.

La funcion recorre el arreglo ingresado para luego regresar la evaluacion correspondiente a la comparacion realizada.

**Parameters**

<i>lista</i>	Secuencia de caracteres ingresados por el usuario.
<i>contador</i>	Variable de control para el contador que recorre el arreglo

**Warning**

El primer dato de la lista es el nombre del ejecutable.

**3.1.2.2 gap()**

```
int blastobject::gap (  
    char ** lista,  
    int n )
```

Compara las secuencias de caracteres con la referencia y cuenta sus vacios.

La funcion recorre el arreglo ingresado para luego regresar la evaluacion correspondiente a la cantidad de vacios existentes en la secuencia comparada.

**Parameters**

<i>lista</i>	Secuencia de caracteres ingresados por el usuario.
<i>n</i>	Variable de control para la cantidad de pasos de recorrido.

**Warning**

El primer dato de la lista es el nombre del ejecutable.

Here is the call graph for this function:

**3.1.2.3 gapclear()**

```
int blastobject::gapclear (  
    char ** lista )
```

Retorna una secuencia con solo los caracteres similares.

La funcion recorre el arreglo ingresado para luego imprimir la secuencia con los caracteres similares y los vacios rellenos con el simbolo de suma +.

**Parameters**

<i>lista</i>	Secuencia de caracteres ingresados por el usuario.
--------------	--

**Warning**

El primer dato de la lista es el nombre del ejecutable.

Here is the caller graph for this function:

**3.1.2.4 raw()**

```
int blastobject::raw (  
    char ** lista )
```

Compara las secuencias de caracteres con la referencia y cuenta su puntuacion cruda.

La funcion recorre el arreglo ingresado para luego regresar la evaluacion correspondiente a la cantidad de puntuacion obtenida con respecto a la referencia.

#### Parameters

<i>lista</i>	Secuencia de caracteres ingresados por el usuario.
--------------	--

#### Warning

El primer dato de la lista es el nombre del ejecutable.

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

- blastobject.h
- blastobject.cpp

## Chapter 4

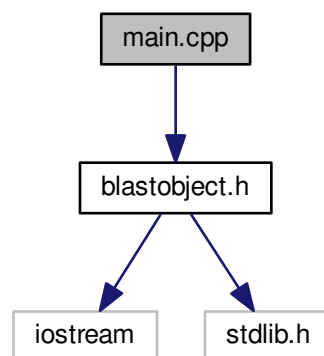
# File Documentation

### 4.1 main.cpp File Reference

En este programa se utiliza el algoritmo BLAST para comparacion de secuencias en su version mas simple, comparando con una base de datos creada por el usuario con el proposito de observar su comportamiento de algoritmo y sus resultados. En este caso dichos objetos son del tipo blastobject.

```
#include "blastobject.h"
```

Include dependency graph for main.cpp:



### Functions

- int **main** (int argc, char \*\*argv)

#### 4.1.1 Detailed Description

En este programa se utiliza el algoritmo BLAST para comparacion de secuencias en su version mas simple, comparando con una base de datos creada por el usuario con el proposito de observar su comportamiento de algoritmo y sus resultados. En este caso dichos objetos son del tipo blastobject.

##### Author

Jose Fernando Gonzalez Salas, Isaac Gomez Sanchez & Alonso Rojas Fernandez.

##### Date

20 de setiembre, 2016

# Index

- blastobject, [5](#)
  - blastobject, [5](#)
  - comparador, [6](#)
  - gap, [6](#)
  - gapclear, [7](#)
  - raw, [7](#)
- comparador
  - blastobject, [6](#)
- gap
  - blastobject, [6](#)
- gapclear
  - blastobject, [7](#)
- main.cpp, [9](#)
- raw
  - blastobject, [7](#)