

FMWS

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	FuzzyMatch Namespace Reference . . . . .	7
4.1.1	Function Documentation . . . . .	7
4.1.1.1	Run() . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	Rank Struct Reference . . . . .	9
5.1.1	Detailed Description . . . . .	10
5.1.2	Constructor & Destructor Documentation . . . . .	10
5.1.2.1	Rank() [1/2] . . . . .	10
5.1.2.2	Rank() [2/2] . . . . .	10
5.1.3	Member Function Documentation . . . . .	10
5.1.3.1	operator>() . . . . .	11
5.1.3.2	operator<() . . . . .	11
5.1.4	Member Data Documentation . . . . .	11
5.1.4.1	id . . . . .	11

5.1.4.2	substring	11
5.1.4.3	levenshtein	11
5.1.4.4	sorensen	12
5.1.4.5	score	12
5.2	Sequence Struct Reference	12
5.2.1	Detailed Description	13
5.2.2	Constructor & Destructor Documentation	13
5.2.2.1	Sequence()	13
5.2.3	Member Data Documentation	13
5.2.3.1	DNA	13
5.2.3.2	id	13
<b>6</b>	<b>File Documentation</b>	<b>15</b>
6.1	include/Driver.hpp File Reference	15
6.1.1	Detailed Description	16
6.1.2	Variable Documentation	16
6.1.2.1	ENABLE_PARALLELIZATION	16
6.2	include/FASTA.hpp File Reference	17
6.2.1	Detailed Description	18
6.2.2	Function Documentation	18
6.2.2.1	ReadFASTA()	18
6.3	include/Levenshtein.hpp File Reference	18
6.3.1	Detailed Description	19
6.3.2	Function Documentation	19
6.3.2.1	LevenshteinDistance()	19
6.4	include/Rank.hpp File Reference	20
6.4.1	Detailed Description	20
6.5	include/Sequence.hpp File Reference	21
6.5.1	Detailed Description	22
6.5.2	Function Documentation	22
6.5.2.1	GetSubstrings()	22

6.6	include/SorensenDice.hpp File Reference	22
6.6.1	Detailed Description	23
6.6.2	Function Documentation	23
6.6.2.1	SorensenDice()	23
6.7	include/Types.hpp File Reference	24
6.7.1	Detailed Description	24
6.7.2	Typedef Documentation	24
6.7.2.1	Index	25
6.8	src/Driver.cpp File Reference	25
6.8.1	Detailed Description	25
6.8.2	Function Documentation	25
6.8.2.1	main()	25
6.9	test/TestBenchmark.cpp File Reference	26
6.9.1	Function Documentation	26
6.9.1.1	TEST() [1/4]	26
6.9.1.2	TEST() [2/4]	27
6.9.1.3	TEST() [3/4]	27
6.9.1.4	TEST() [4/4]	27
6.10	test/TestFASTA.cpp File Reference	27
6.10.1	Function Documentation	28
6.10.1.1	TEST()	28
6.11	test/TestLevenshtein.cpp File Reference	29
6.11.1	Function Documentation	29
6.11.1.1	TEST() [1/2]	29
6.11.1.2	TEST() [2/2]	30
6.12	test/TestSequence.cpp File Reference	30
6.12.1	Function Documentation	30
6.12.1.1	TEST()	31
6.13	test/TestSorensen.cpp File Reference	31
6.13.1	Function Documentation	31
6.13.1.1	TEST()	31



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">FuzzyMatch</a> . . . . .	7
--------------------------------------	---





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Rank</a>	.....	9
<a href="#">Sequence</a>	.....	12



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

include/ <a href="#">Driver.hpp</a>	
Holds main logic of code base . . . . .	15
include/ <a href="#">FASTA.hpp</a>	
Handle reading in FASTA files into sequences . . . . .	17
include/ <a href="#">Levenshtein.hpp</a>	
Calculate Levenshtein distance . . . . .	18
include/ <a href="#">Rank.hpp</a>	
Defines how we rank the best substrings . . . . .	20
include/ <a href="#">Sequence.hpp</a>	
Provides functionality for manipulating DNA sequences . . . . .	21
include/ <a href="#">SorensenDice.hpp</a>	
Calculate Sorensen Dice coefficient . . . . .	22
include/ <a href="#">Types.hpp</a>	
Defines various custom data types for ease-of-use . . . . .	24
src/ <a href="#">Driver.cpp</a>	
Serves as the command-line interface . . . . .	25
test/ <a href="#">TestBenchmark.cpp</a>	26
test/ <a href="#">TestFASTA.cpp</a>	27
test/ <a href="#">TestLevenshtein.cpp</a>	29
test/ <a href="#">TestSequence.cpp</a>	30
test/ <a href="#">TestSorensen.cpp</a>	31



## Chapter 4

# Namespace Documentation

### 4.1 FuzzyMatch Namespace Reference

#### Functions

- void [Run](#) (const std::string &filePath, const std::string &substring, const bool verbose=(true))  
*Run the program, called from CLI.*

#### 4.1.1 Function Documentation

##### 4.1.1.1 Run()

```
void FuzzyMatch::Run (  
    const std::string & filePath,  
    const std::string & substring,  
    const bool verbose = (true) ) [inline]
```

Run the program, called from CLI.

Definition at line 25 of file Driver.hpp.



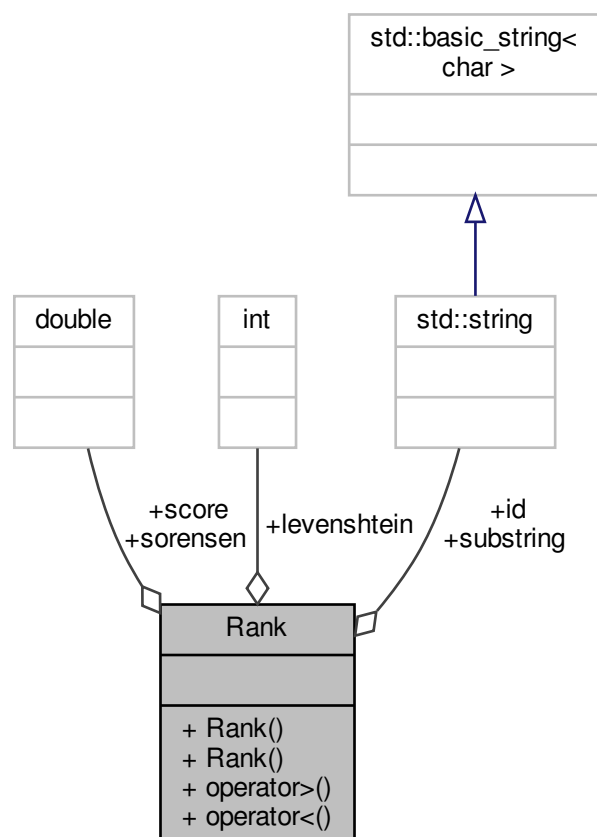
## Chapter 5

# Class Documentation

### 5.1 Rank Struct Reference

```
#include <Rank.hpp>
```

Collaboration diagram for Rank:



## Public Member Functions

- [Rank](#) ()
- [Rank](#) (const std::string &seqID, const std::string &substr, const int lScore, const double sScore)
- bool [operator>](#) (const [Rank](#) &r) const
- bool [operator<](#) (const [Rank](#) &r) const

## Public Attributes

- std::string [id](#)
- std::string [substring](#)
- int [levenshtein](#)
- double [sorensen](#)
- double [score](#) = 100000

### 5.1.1 Detailed Description

Definition at line 11 of file Rank.hpp.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 [Rank\(\)](#) [1/2]

```
Rank::Rank ( ) [inline], [explicit]
```

Definition at line 19 of file Rank.hpp.

#### 5.1.2.2 [Rank\(\)](#) [2/2]

```
Rank::Rank (
    const std::string & seqID,
    const std::string & substr,
    const int lScore,
    const double sScore ) [inline], [explicit]
```

Definition at line 20 of file Rank.hpp.

### 5.1.3 Member Function Documentation



#### 5.1.3.1 `operator>()`

```
bool Rank::operator> (
    const Rank & r ) const [inline]
```

Definition at line 32 of file Rank.hpp.

#### 5.1.3.2 `operator<()`

```
bool Rank::operator< (
    const Rank & r ) const [inline]
```

Definition at line 35 of file Rank.hpp.

### 5.1.4 Member Data Documentation

#### 5.1.4.1 `id`

```
std::string Rank::id
```

Definition at line 12 of file Rank.hpp.

#### 5.1.4.2 `substring`

```
std::string Rank::substring
```

Definition at line 13 of file Rank.hpp.

#### 5.1.4.3 `levenshtein`

```
int Rank::levenshtein
```

Definition at line 14 of file Rank.hpp.

#### 5.1.4.4 sorensen

```
double Rank::sorensen
```

Definition at line 15 of file Rank.hpp.

#### 5.1.4.5 score

```
double Rank::score = 100000
```

Definition at line 16 of file Rank.hpp.

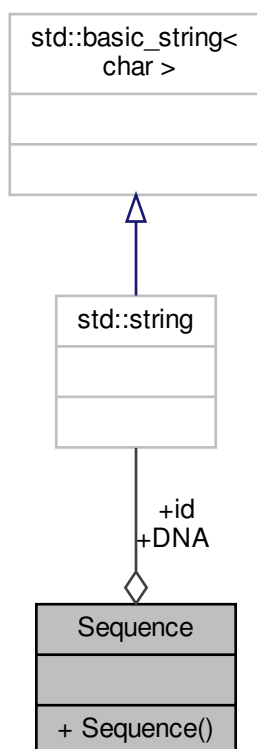
The documentation for this struct was generated from the following file:

- [include/Rank.hpp](#)

## 5.2 Sequence Struct Reference

```
#include <Sequence.hpp>
```

Collaboration diagram for Sequence:



## Public Member Functions

- [Sequence](#) (const std::string &seqID, const std::string &seq)

## Public Attributes

- std::string [DNA](#)
- std::string [id](#)

### 5.2.1 Detailed Description

Definition at line 13 of file Sequence.hpp.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 Sequence()

```
Sequence::Sequence (
    const std::string & seqID,
    const std::string & seq ) [inline], [explicit]
```

Definition at line 17 of file Sequence.hpp.

### 5.2.3 Member Data Documentation

#### 5.2.3.1 DNA

```
std::string Sequence::DNA
```

Definition at line 14 of file Sequence.hpp.

#### 5.2.3.2 id

```
std::string Sequence::id
```

Definition at line 15 of file Sequence.hpp.

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

- include/[Sequence.hpp](#)



## Chapter 6

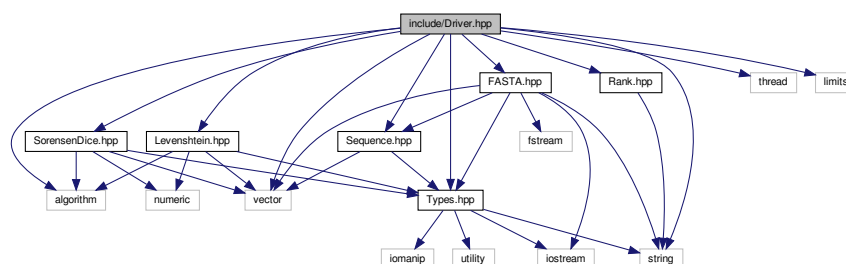
# File Documentation

### 6.1 include/Driver.hpp File Reference

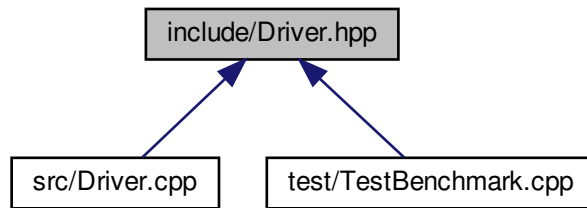
Holds main logic of code base.

```
#include <algorithm>
#include <string>
#include <vector>
#include <thread>
#include <limits>
#include "FASTA.hpp"
#include "Levenshtein.hpp"
#include "SorensenDice.hpp"
#include "Rank.hpp"
#include "Sequence.hpp"
#include "Types.hpp"
```

Include dependency graph for Driver.hpp:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [FuzzyMatch](#)

## Functions

- void [FuzzyMatch::Run](#) (const std::string &filePath, const std::string &substring, const bool verbose=(true))  
*Run the program, called from CLI.*

## Variables

- constexpr bool [ENABLE\\_PARALLELIZATION](#) = true

### 6.1.1 Detailed Description

Holds main logic of code base.

### 6.1.2 Variable Documentation

#### 6.1.2.1 ENABLE\_PARALLELIZATION

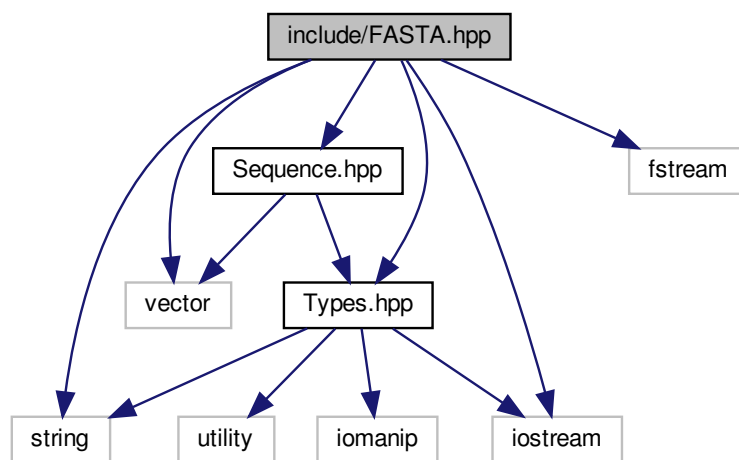
```
constexpr bool ENABLE_PARALLELIZATION = true
```

Definition at line 18 of file Driver.hpp.

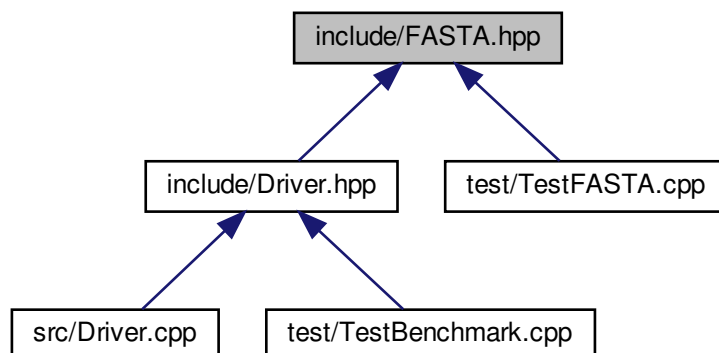
## 6.2 include/FASTA.hpp File Reference

Handle reading in FASTA files into sequences.

```
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include "Types.hpp"
#include "Sequence.hpp"
Include dependency graph for FASTA.hpp:
```



This graph shows which files directly or indirectly include this file:



## Functions

- `std::vector< Sequence > ReadFASTA` (`const std::string &filePath`)  
*Read in a simple FASTA file and return a vector of sequences.*

### 6.2.1 Detailed Description

Handle reading in FASTA files into sequences.

### 6.2.2 Function Documentation

#### 6.2.2.1 ReadFASTA()

```
std::vector<Sequence> ReadFASTA (  
    const std::string & filePath ) [inline]
```

Read in a simple FASTA file and return a vector of sequences.

#### Note

C++11 and higher doesn't return a copy of the data here, it performs a move operation. This is important when dealing with the human genome.

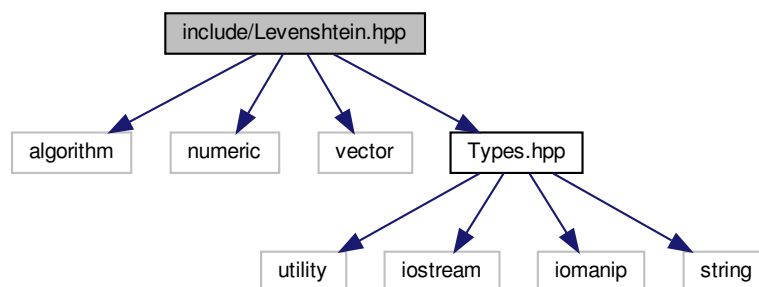
Definition at line 23 of file FASTA.hpp.

## 6.3 include/Levenshtein.hpp File Reference

Calculate Levenshtein distance.

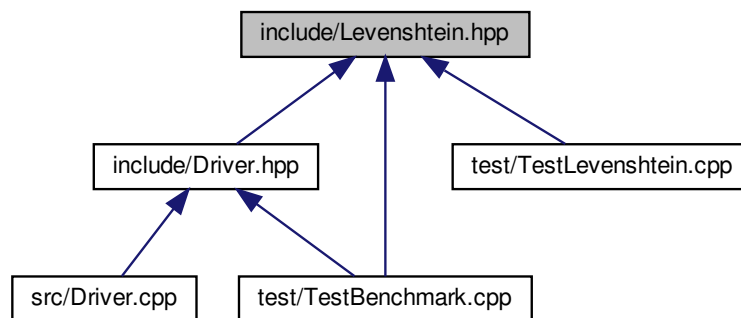
```
#include <algorithm>  
#include <numeric>  
#include <vector>  
#include "Types.hpp"
```

Include dependency graph for Levenshtein.hpp:





This graph shows which files directly or indirectly include this file:



## Functions

- int [LevenshteinDistance](#) (const std::string &seqA, const std::string &seqB)  
*Calculate Levenshtein distance on two sequences.*

### 6.3.1 Detailed Description

Calculate Levenshtein distance.

### 6.3.2 Function Documentation

#### 6.3.2.1 LevenshteinDistance()

```
int LevenshteinDistance (  
    const std::string & seqA,  
    const std::string & seqB ) [inline]
```

Calculate Levenshtein distance on two sequences.

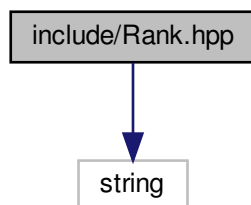
Definition at line 18 of file Levenshtein.hpp.

## 6.4 include/Rank.hpp File Reference

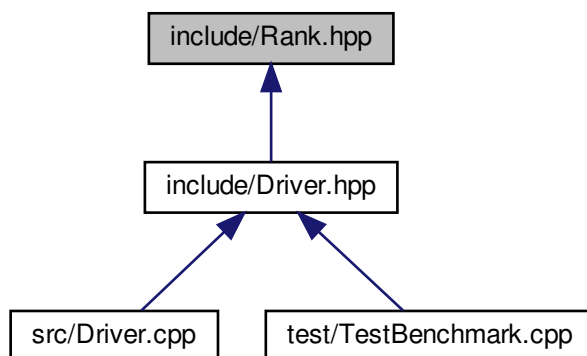
Defines how we rank the best substrings.

```
#include <string>
```

Include dependency graph for Rank.hpp:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rank](#)

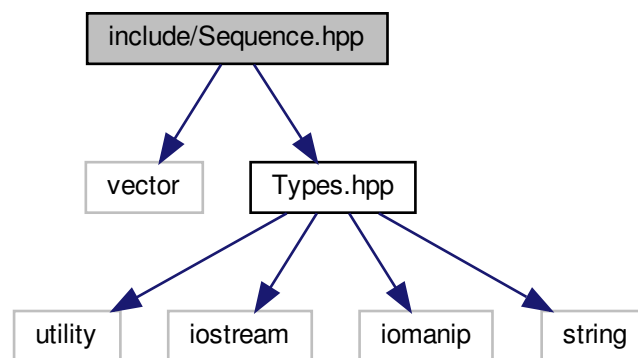
#### 6.4.1 Detailed Description

Defines how we rank the best substrings.

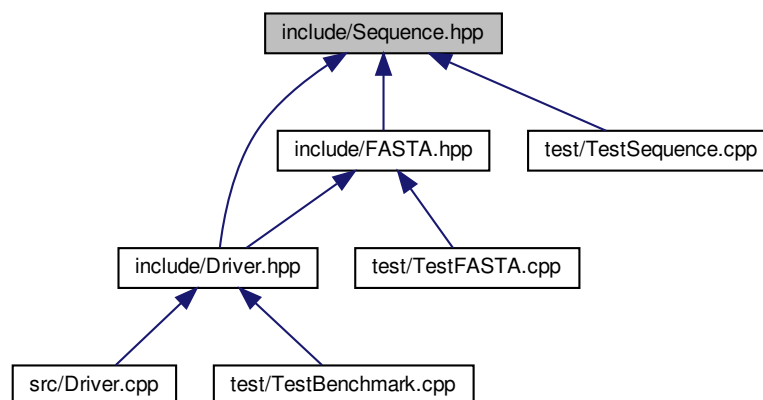
## 6.5 include/Sequence.hpp File Reference

Provides functionality for manipulating DNA sequences.

```
#include <vector>
#include "Types.hpp"
Include dependency graph for Sequence.hpp:
```



This graph shows which files directly or indirectly include this file:



### Classes

- struct [Sequence](#)

## Functions

- `std::vector< std::string > GetSubstrings (const Index start, const Index length, Index &amount, const std::string &sequence)`

*Extract substrings from given string AKA make k-mers.*

### 6.5.1 Detailed Description

Provides functionality for manipulating DNA sequences.

### 6.5.2 Function Documentation

#### 6.5.2.1 GetSubstrings()

```
std::vector<std::string> GetSubstrings (
    const Index start,
    const Index length,
    Index & amount,
    const std::string & sequence ) [inline]
```

Extract substrings from given string AKA make k-mers.

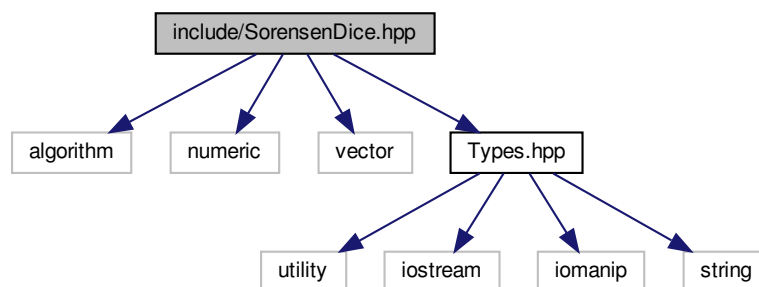
Definition at line 25 of file Sequence.hpp.

## 6.6 include/SorensenDice.hpp File Reference

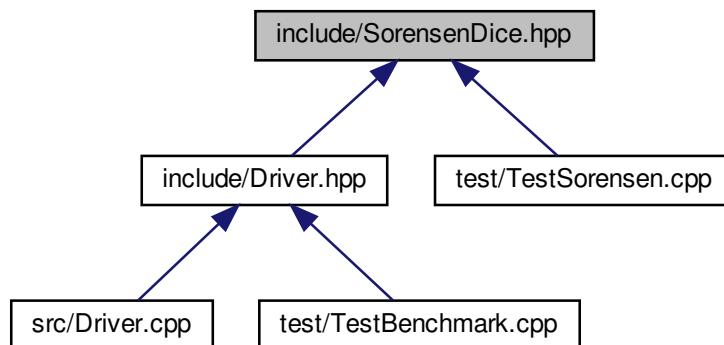
Calculate Sorensen Dice coefficient.

```
#include <algorithm>
#include <numeric>
#include <vector>
#include "Types.hpp"
```

Include dependency graph for SorensenDice.hpp:



This graph shows which files directly or indirectly include this file:



## Functions

- double [SorensenDice](#) (const std::string &seqA, const std::string &seqB)  
*Calculate Sorensen Dice distance.*

### 6.6.1 Detailed Description

Calculate Sorensen Dice coefficient.

### 6.6.2 Function Documentation

#### 6.6.2.1 SorensenDice()

```
double SorensenDice (  
    const std::string & seqA,  
    const std::string & seqB ) [inline]
```

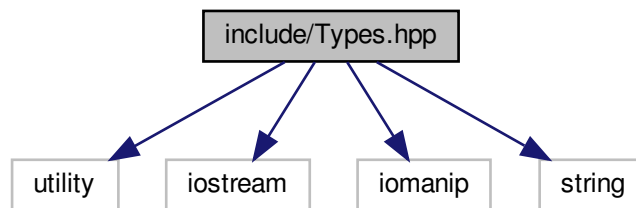
Calculate Sorensen Dice distance.

Definition at line 17 of file SorensenDice.hpp.

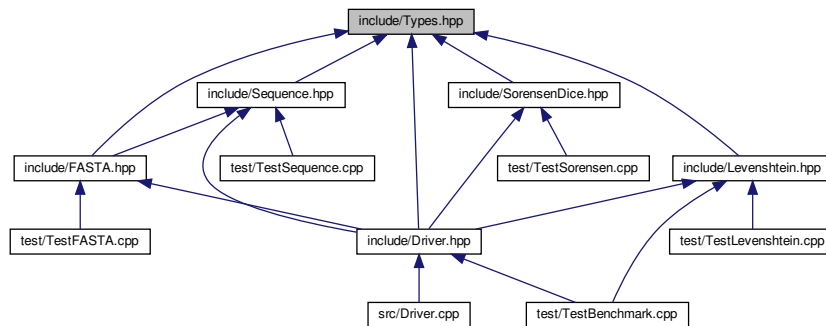
## 6.7 include/Types.hpp File Reference

Defines various custom data types for ease-of-use.

```
#include <utility>
#include <iostream>
#include <iomanip>
#include <string>
Include dependency graph for Types.hpp:
```



This graph shows which files directly or indirectly include this file:



### Typedefs

- using [Index](#) = std::size\_t

#### 6.7.1 Detailed Description

Defines various custom data types for ease-of-use.

#### 6.7.2 Typedef Documentation

### 6.7.2.1 Index

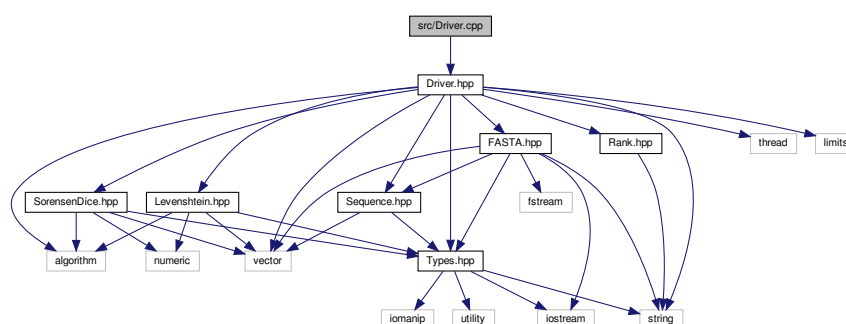
```
using Index = std::size_t
```

Definition at line 21 of file Types.hpp.

## 6.8 src/Driver.cpp File Reference

Serves as the command-line interface.

```
#include "Driver.hpp"
Include dependency graph for Driver.cpp:
```



## Functions

- int [main](#) (int argc, char \*argv[ ])

### 6.8.1 Detailed Description

Serves as the command-line interface.

### 6.8.2 Function Documentation

#### 6.8.2.1 main()

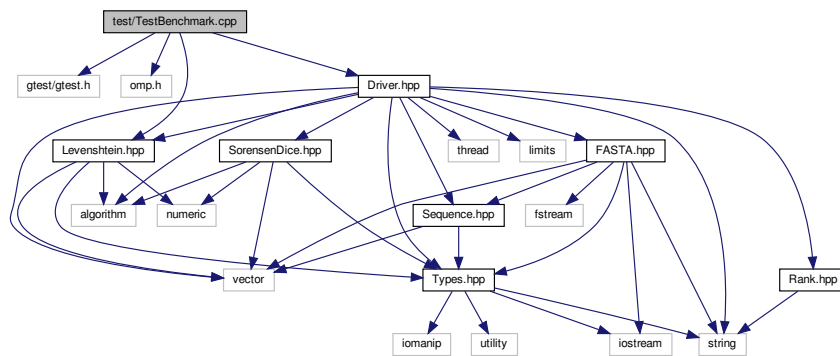
```
int main (
    int argc,
    char * argv[ ] )
```

Definition at line 9 of file Driver.cpp.

## 6.9 test/TestBenchmark.cpp File Reference

```
#include "gtest/gtest.h"
#include "omp.h"
#include "Driver.hpp"
#include "Levenshtein.hpp"
```

Include dependency graph for TestBenchmark.cpp:



### Functions

- **TEST** (Benchmark, EColi)  
*E. coli benchmark.*
- **TEST** (Benchmark, BacAnth)  
*Bacillus Anthracis benchmark.*
- **TEST** (Benchmark, BacCer)  
*Bacillus Cereus benchmark.*
- **TEST** (Benchmark, HomoSap)  
*Homo Sapien benchmark.*

### 6.9.1 Function Documentation

#### 6.9.1.1 TEST() [1/4]

```
TEST (
    Benchmark ,
    EColi )
```

*E. coli benchmark.*

Definition at line 9 of file TestBenchmark.cpp.



### 6.9.1.2 TEST() [2/4]

```
TEST (
    Benchmark ,
    BacAnth )
```

Bacillus Anthracis benchmark.

Definition at line 24 of file TestBenchmark.cpp.

### 6.9.1.3 TEST() [3/4]

```
TEST (
    Benchmark ,
    BacCer )
```

Bacillus Cereus benchmark.

Definition at line 38 of file TestBenchmark.cpp.

### 6.9.1.4 TEST() [4/4]

```
TEST (
    Benchmark ,
    HomoSap )
```

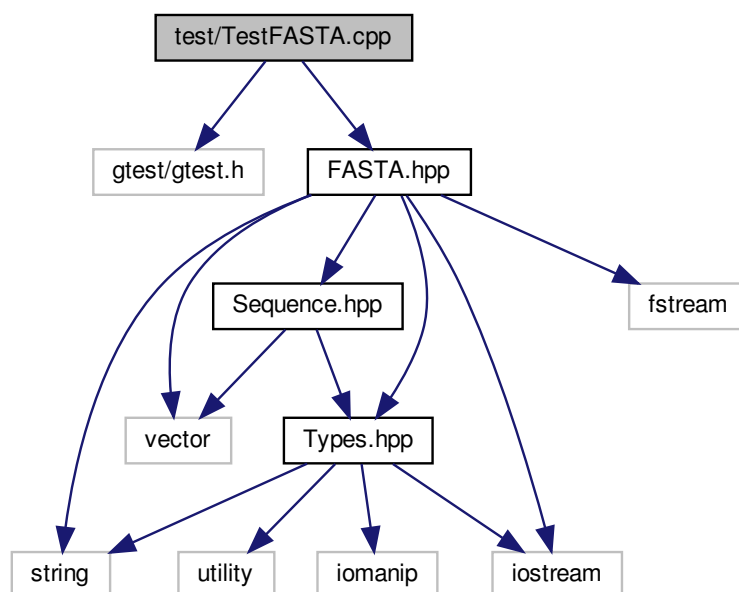
Homo Sapien benchmark.

Definition at line 52 of file TestBenchmark.cpp.

## 6.10 test/TestFASTA.cpp File Reference

```
#include "gtest/gtest.h"
#include "FASTA.hpp"
```

Include dependency graph for TestFASTA.cpp:



## Functions

- [TEST](#) (FASTA, Small)  
*Read in a small, simple FASTA file.*

### 6.10.1 Function Documentation

#### 6.10.1.1 TEST()

```
TEST (
    FASTA ,
    Small )
```

Read in a small, simple FASTA file.

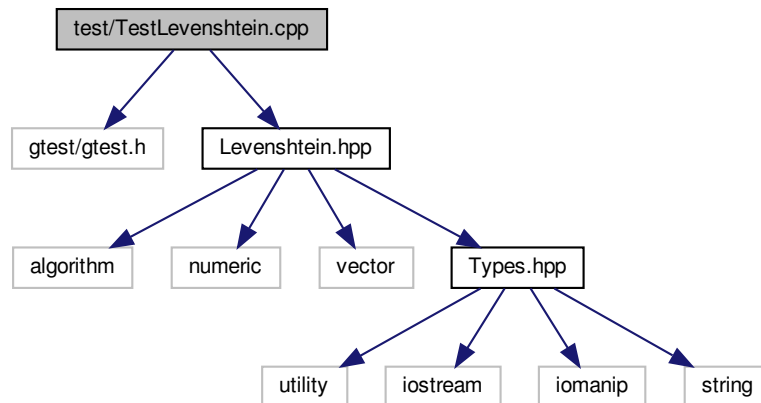
Definition at line 8 of file TestFASTA.cpp.

## 6.11 test/TestLevenshtein.cpp File Reference

```
#include "gtest/gtest.h"
```

```
#include "Levenshtein.hpp"
```

Include dependency graph for TestLevenshtein.cpp:



### Functions

- [TEST](#) (Levenshtein, SmallEqual)  
*Small equal-sized sequences.*
- [TEST](#) (Levenshtein, SmallDifferent)  
*Small different-sized sequences.*

#### 6.11.1 Function Documentation

##### 6.11.1.1 TEST() [1/2]

```
TEST (
    Levenshtein ,
    SmallEqual )
```

Small equal-sized sequences.

Definition at line 8 of file `TestLevenshtein.cpp`.

### 6.11.1.2 TEST() [2/2]

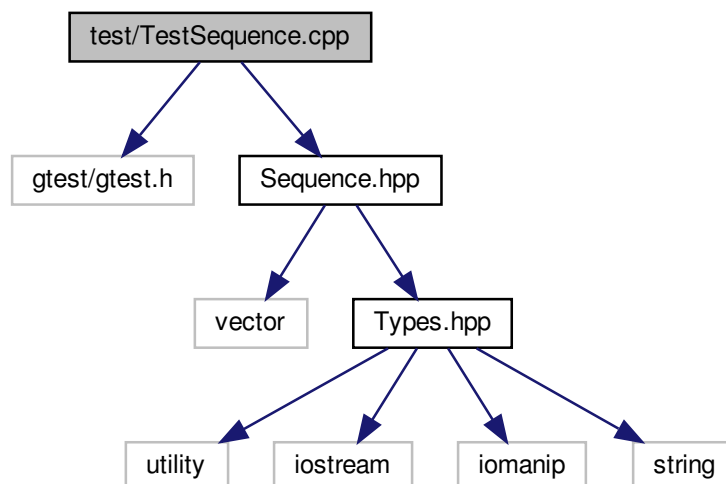
```
TEST (
    Levenshtein ,
    SmallDifferent )
```

Small different-sized sequences.

Definition at line 17 of file TestLevenshtein.cpp.

## 6.12 test/TestSequence.cpp File Reference

```
#include "gtest/gtest.h"
#include "Sequence.hpp"
Include dependency graph for TestSequence.cpp:
```



### Functions

- [TEST](#) ([Sequence](#), Substr)  
*Generate a few small k-mers.*

### 6.12.1 Function Documentation

## 6.12.1.1 TEST()

```
TEST (
    Sequence ,
    Substr )
```

Generate a few small k-mers.

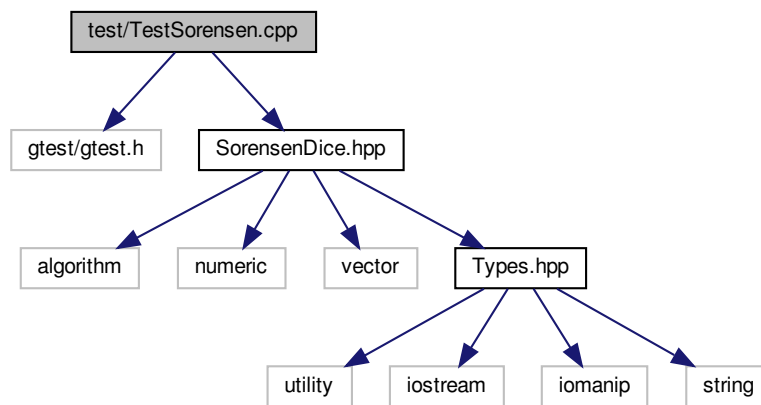
Definition at line 8 of file TestSequence.cpp.

## 6.13 test/TestSorensen.cpp File Reference

```
#include "gtest/gtest.h"
```

```
#include "SorensenDice.hpp"
```

Include dependency graph for TestSorensen.cpp:



## Functions

- **TEST** ([SorensenDice](#), SmallEqual)

*Small equal-sized sequences.*

## 6.13.1 Function Documentation

## 6.13.1.1 TEST()

```
TEST (
    SorensenDice ,
    SmallEqual )
```

Small equal-sized sequences.

Definition at line 8 of file TestSorensen.cpp.



# Index

- DNA
  - Sequence, [13](#)
- Driver.cpp
  - main, [25](#)
- Driver.hpp
  - ENABLE\_PARALLELIZATION, [16](#)
- ENABLE\_PARALLELIZATION
  - Driver.hpp, [16](#)
- FASTA.hpp
  - ReadFASTA, [18](#)
- FuzzyMatch, [7](#)
  - Run, [7](#)
- GetSubstrings
  - Sequence.hpp, [22](#)
- id
  - Rank, [11](#)
  - Sequence, [13](#)
- include/Driver.hpp, [15](#)
- include/FASTA.hpp, [17](#)
- include/Levenshtein.hpp, [18](#)
- include/Rank.hpp, [20](#)
- include/Sequence.hpp, [21](#)
- include/SorensenDice.hpp, [22](#)
- include/Types.hpp, [24](#)
- Index
  - Types.hpp, [24](#)
- levenshtein
  - Rank, [11](#)
- Levenshtein.hpp
  - LevenshteinDistance, [19](#)
- LevenshteinDistance
  - Levenshtein.hpp, [19](#)
- main
  - Driver.cpp, [25](#)
- operator<
  - Rank, [11](#)
- operator>
  - Rank, [10](#)
- Rank, [9](#)
  - id, [11](#)
  - levenshtein, [11](#)
  - operator<, [11](#)
  - operator>, [10](#)
  - Rank, [10](#)
  - score, [12](#)
  - sorensen, [11](#)
  - substring, [11](#)
- ReadFASTA
  - FASTA.hpp, [18](#)
- Run
  - FuzzyMatch, [7](#)
- score
  - Rank, [12](#)
- Sequence, [12](#)
  - DNA, [13](#)
  - id, [13](#)
  - Sequence, [13](#)
- Sequence.hpp
  - GetSubstrings, [22](#)
- sorensen
  - Rank, [11](#)
- SorensenDice
  - SorensenDice.hpp, [23](#)
- SorensenDice.hpp
  - SorensenDice, [23](#)
- src/Driver.cpp, [25](#)
- substring
  - Rank, [11](#)
- TEST
  - TestBenchmark.cpp, [26, 27](#)
  - TestFASTA.cpp, [28](#)
  - TestLevenshtein.cpp, [29](#)
  - TestSequence.cpp, [30](#)
  - TestSorensen.cpp, [31](#)
- test/TestBenchmark.cpp, [26](#)
- test/TestFASTA.cpp, [27](#)
- test/TestLevenshtein.cpp, [29](#)
- test/TestSequence.cpp, [30](#)
- test/TestSorensen.cpp, [31](#)
- TestBenchmark.cpp
  - TEST, [26, 27](#)
- TestFASTA.cpp
  - TEST, [28](#)
- TestLevenshtein.cpp
  - TEST, [29](#)
- TestSequence.cpp
  - TEST, [30](#)
- TestSorensen.cpp
  - TEST, [31](#)
- Types.hpp
  - Index, [24](#)