

# Prediction medicale

1.0

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	pile Struct Reference . . . . .	5
3.2	Pile Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	5
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	contextmedical.c File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Function Documentation . . . . .	7
4.1.2.1	conversionchar() . . . . .	7
4.2	contextmedical.h File Reference . . . . .	8
4.2.1	Detailed Description . . . . .	8
4.2.2	Function Documentation . . . . .	8
4.2.2.1	conversionchar() . . . . .	8
4.3	main.c File Reference . . . . .	9
4.3.1	Detailed Description . . . . .	9
4.4	patient.c File Reference . . . . .	10
4.4.1	Detailed Description . . . . .	10
4.4.2	Function Documentation . . . . .	10

4.4.2.1	<code>creermotif()</code>	10
4.4.2.2	<code>limitermotif()</code>	11
4.5	<code>patient.h</code> File Reference	11
4.5.1	Detailed Description	11
4.5.2	Function Documentation	12
4.5.2.1	<code>creermotif()</code>	12
4.5.2.2	<code>limitermotif()</code>	12
4.6	<code>pile.c</code> File Reference	12
4.6.1	Detailed Description	13
4.6.2	Function Documentation	13
4.6.2.1	<code>depiler()</code>	13
4.6.2.2	<code>empiler()</code>	14
4.6.2.3	<code>estvide()</code>	14
4.6.2.4	<code>existe()</code>	14
4.6.2.5	<code>vider()</code>	15
4.6.2.6	<code>visualiser()</code>	15
4.7	<code>pile.h</code> File Reference	15
4.7.1	Detailed Description	16
4.7.2	Function Documentation	16
4.7.2.1	<code>depiler()</code>	16
4.7.2.2	<code>empiler()</code>	16
4.7.2.3	<code>estvide()</code>	17
4.7.2.4	<code>existe()</code>	17
4.7.2.5	<code>vider()</code>	17
4.7.2.6	<code>visualiser()</code>	18
4.8	<code>prediction.c</code> File Reference	18
4.8.1	Detailed Description	18
4.8.2	Function Documentation	19
4.8.2.1	<code>conversionint()</code>	19
4.8.2.2	<code>conversionmotif()</code>	19

4.8.2.3	<a href="#">conversionmotif()</a>	20
4.8.2.4	<a href="#">prediction()</a>	20
4.8.2.5	<a href="#">score()</a>	20
4.9	<a href="#">prediction.h File Reference</a>	21
4.9.1	<a href="#">Detailed Description</a>	21
4.9.2	<a href="#">Function Documentation</a>	21
4.9.2.1	<a href="#">conversionint()</a>	21
4.9.2.2	<a href="#">conversionmotif()</a>	22
4.9.2.3	<a href="#">conversiontmotif()</a>	22
4.9.2.4	<a href="#">prediction()</a>	22
4.9.2.5	<a href="#">score()</a>	23
4.10	<a href="#">predictionpartielle.c File Reference</a>	23
4.10.1	<a href="#">Detailed Description</a>	24
4.10.2	<a href="#">Function Documentation</a>	24
4.10.2.1	<a href="#">calculerProba1()</a>	24
4.10.2.2	<a href="#">calculerProba2()</a>	25
4.10.2.3	<a href="#">decoupermotif()</a>	25
4.10.2.4	<a href="#">predictionpartielle()</a>	25
4.10.2.5	<a href="#">remplirListePredictions1()</a>	27
4.10.2.6	<a href="#">remplirListePredictions2()</a>	27
4.10.2.7	<a href="#">scoremax()</a>	28
4.11	<a href="#">predictionpartielle.h File Reference</a>	28
4.11.1	<a href="#">Detailed Description</a>	29
4.11.2	<a href="#">Function Documentation</a>	29
4.11.2.1	<a href="#">calculerProba1()</a>	29
4.11.2.2	<a href="#">calculerProba2()</a>	30
4.11.2.3	<a href="#">decoupermotif()</a>	30
4.11.2.4	<a href="#">predictionpartielle()</a>	30
4.11.2.5	<a href="#">remplirListePredictions1()</a>	31
4.11.2.6	<a href="#">remplirListePredictions2()</a>	31
4.11.2.7	<a href="#">scoremax()</a>	32



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">pile</a>	.....	5
<a href="#">Pile</a>		
<a href="#">Pile</a> contenant les predictions	.....	5





## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">contextmedical.c</a>	Contexte medical dans lequel on souhaite realiser la prediction . . . . .	7
<a href="#">contextmedical.h</a>	Contexte medical dans lequel on souhaite realiser la prediction . . . . .	8
<a href="#">main.c</a>	Fonction principale du programme . . . . .	9
<a href="#">patient.c</a>	Antecedents du patient (motif) . . . . .	10
<a href="#">patient.h</a>	Antecedents du patient (motif) . . . . .	11
<a href="#">pile.c</a>	Structure de pile. Permet de creer une liste de predictions . . . . .	12
<a href="#">pile.h</a>	Structure de pile. Permet de creer une liste de predictions . . . . .	15
<a href="#">prediction.c</a>	Prediction . . . . .	18
<a href="#">prediction.h</a>	Prediction des suites d'un patient a partir de l'analyse des motifs de patients contenus dans une base sequentielle . . . . .	21
<a href="#">predictionpartielle.c</a>	Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle) . . . . .	23
<a href="#">predictionpartielle.h</a>	Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle) . . . . .	28



## Chapter 3

# Data Structure Documentation

### 3.1 pile Struct Reference

#### Data Fields

- int **valeur**
- struct [pile](#) \* **prec**

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

- [pile.h](#)

### 3.2 Pile Struct Reference

[Pile](#) contenant les predictions.

```
#include <pile.h>
```

#### 3.2.1 Detailed Description

[Pile](#) contenant les predictions.

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

- [pile.h](#)



## Chapter 4

# File Documentation

### 4.1 contextemedical.c File Reference

Contexte medical dans lequel on souhaite realiser la prediction.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "contextemedical.h"
```

#### Functions

- int [conversionchar](#) (char a)

*Fonction de conversion d'un caractere en l'entier qu'il represente. Utile pour convertir le contexte medical (ecrit sous forme de caractere dans la console) en entier.*

#### 4.1.1 Detailed Description

Contexte medical dans lequel on souhaite realiser la prediction.

##### Author

Stella Zevio

##### Version

1.0

##### Date

2015-06-20

Prediction medicale

#### 4.1.2 Function Documentation

##### 4.1.2.1 conversionchar()

```
int conversionchar (
    char a )
```

Fonction de conversion d'un caractere en l'entier qu'il represente. Utile pour convertir le contexte medical (ecrit sous forme de caractere dans la console) en entier.

### Parameters

<i>a</i>	- le caractere a convertir
----------	----------------------------

### Returns

aint - l'entier represente par a

## 4.2 contextemedical.h File Reference

Contexte medical dans lequel on souhaite realiser la prediction.

### Functions

- int [conversionchar](#) (char a)

*Fonction de conversion d'un caractere en l'entier qu'il represente. Utile pour convertir le contexte medical (ecrit sous forme de caractere dans la console) en entier.*

### 4.2.1 Detailed Description

Contexte medical dans lequel on souhaite realiser la prediction.

### Author

Stella Zevio

### Version

1.0

### Date

2015-06-20

Prediction medicale

## 4.2.2 Function Documentation

### 4.2.2.1 conversionchar()

```
int conversionchar (  
    char a )
```

Fonction de conversion d'un caractere en l'entier qu'il represente. Utile pour convertir le contexte medical (ecrit sous forme de caractere dans la console) en entier.

#### Parameters

<i>a</i>	- le caractere a convertir
----------	----------------------------

#### Returns

aint - l'entier represente par a

## 4.3 main.c File Reference

Fonction principale du programme.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "patient.h"
#include "contextemedical.h"
#include "prediction.h"
#include "predictionpartielle.h"
```

#### Functions

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

### 4.3.1 Detailed Description

Fonction principale du programme.

#### Author

Stella Zevio

#### Version

1.0

#### Date

2015-06-20

Prediction medicale

## 4.4 patient.c File Reference

Antecedents du patient (motif)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "patient.h"
```

### Functions

- char \* [creermotif](#) (int argc, char \*argv[])  
*Fonction permettant de recuperer le motif d'un patient passe en parametre sur le terminal (ses antecedents)*
- char \* [limitermotif](#) (char \*motif)  
*Fonction permettant de limiter le motif d'un patient a ses trois dernieres hospitalisations, afin de proposer une prediction pertinente.*

### 4.4.1 Detailed Description

Antecedents du patient (motif)

Author

Stella Zevio

Version

1.0

Date

2015-06-20

Prediction medicale

### 4.4.2 Function Documentation

#### 4.4.2.1 creermotif()

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

Fonction permettant de recuperer le motif d'un patient passe en parametre sur le terminal (ses antecedents)



## Parameters

<i>argc</i>	- le nombre de parametres passes au programme
<i>argv</i>	- les parametres passes au programme

## Returns

motif - le motif du patient

## 4.4.2.2 limitermotif()

```
char * limitermotif (
    char * motif )
```

Fonction permettant de limiter le motif d'un patient a ses trois dernieres hospitalisations, afin de proposer une prediction pertinente.

## Parameters

<i>motif</i>	- le motif complet du patient
--------------	-------------------------------

## Returns

motifpred - le motif du patient utilise pour la prediction (les 3 dernieres hospitalisations)

## 4.5 patient.h File Reference

Antecedents du patient (motif)

## Functions

- char \* [creermotif](#) (int argc, char \*argv[])  
*Fonction permettant de recuperer le motif d'un patient passe en parametre sur le terminal (ses antecedents)*
- char \* [limitermotif](#) (char \*motif)  
*Fonction permettant de limiter le motif d'un patient a ses trois dernieres hospitalisations, afin de proposer une prediction pertinente.*

### 4.5.1 Detailed Description

Antecedents du patient (motif)

## Author

Stella Zevio

## Version

1.0

## Date

2015-06-20

Prediction medicale

## 4.5.2 Function Documentation

### 4.5.2.1 creermotif()

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

Fonction permettant de recuperer le motif d'un patient passe en parametre sur le terminal (ses antecedents)

#### Parameters

<i>argc</i>	- le nombre de parametres passes au programme
<i>argv</i>	- les parametres passes au programme

#### Returns

motif - le motif du patient

### 4.5.2.2 limitermotif()

```
char* limitermotif (
    char * motif )
```

Fonction permettant de limiter le motif d'un patient a ses trois dernieres hospitalisations, afin de proposer une prediction pertinente.

#### Parameters

<i>motif</i>	- le motif complet du patient
--------------	-------------------------------

#### Returns

motifpred - le motif du patient utilise pour la prediction (les 3 dernieres hospitalisations)

## 4.6 pile.c File Reference

Structure de pile. Permet de creer une liste de predictions.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "pile.h"
```

## Functions

- void `empiler (pile **p, int valeur)`  
*Fonction permettant d'empiler une valeur.*
- int `depiler (pile **p)`  
*Fonction permettant de depiler une valeur.*
- void `vider (pile **p)`  
*Fonction permettant de vider la pile.*
- void `visualiser (pile *p)`  
*Fonction permettant de visualiser la pile.*
- int `existe (pile *p, int valeur)`  
*Fonction permettant de verifier si une valeur existe dans la pile.*
- int `estvide (pile **p)`  
*Fonction permettant de verifier si une pile est vide.*

### 4.6.1 Detailed Description

Structure de pile. Permet de creer une liste de predictions.

#### Author

Stella Zevio

#### Version

1.0

#### Date

2015-06-20

Prediction medicale

### 4.6.2 Function Documentation

#### 4.6.2.1 depiler()

```
depiler (  
    pile ** p )
```

Fonction permettant de depiler une valeur.

#### Parameters

<i>p</i>	- la pile
----------	-----------

**Returns**

valeur - la valeur qu'on vient de depiler

**4.6.2.2 empiler()**

```
empiler (
    pile ** p,
    int val )
```

Fonction permettant d'empiler une valeur.

**Parameters**

<i>p</i>	- la pile
<i>valeur</i>	- la valeur a empiler

**4.6.2.3 estvide()**

```
estvide (
    pile ** p )
```

Fonction permettant de verifier si une pile est vide.

**Parameters**

<i>p</i>	- la pile
----------	-----------

**Returns**

1 si la pile est vide  
0 si la pile n'est pas vide

**4.6.2.4 existe()**

```
existe (
    pile * p,
    int valeur )
```

Fonction permettant de verifier si une valeur existe dans la pile.

**Parameters**

<i>p</i>	- la pile
<i>valeur</i>	- la valeur dont on veut verifier la presence dans la pile

### Returns

existe - 0 si la valeur n'existe pas dans la pile, 1 sinon

#### 4.6.2.5 vider()

```
vider (
    pile ** p )
```

Fonction permettant de vider la pile.

### Parameters

<i>p</i>	- la pile
----------	-----------

#### 4.6.2.6 visualiser()

```
visualiser (
    pile * p )
```

Fonction permettant de visualiser la pile.

### Parameters

<i>p</i>	- la pile
----------	-----------

## 4.7 pile.h File Reference

Structure de pile. Permet de creer une liste de predictions.

### Data Structures

- struct `pile`

### Typedefs

- typedef struct `pile` **pile**

### Functions

- void `empiler` (`pile **p`, int valeur)  
*Fonction permettant d'empiler une valeur.*
- int `depiler` (`pile **p`)  
*Fonction permettant de depiler une valeur.*

- void `vider (pile **p)`  
*Fonction permettant de vider la pile.*
- void `visualiser (pile *p)`  
*Fonction permettant de visualiser la pile.*
- int `existe (pile *p, int valeur)`  
*Fonction permettant de verifier si une valeur existe dans la pile.*
- int `estvide (pile **p)`  
*Fonction permettant de verifier si une pile est vide.*

#### 4.7.1 Detailed Description

Structure de pile. Permet de creer une liste de predictions.

##### Author

Stella Zevio

##### Version

1.0

##### Date

2015-06-20

Prediction medicale

#### 4.7.2 Function Documentation

##### 4.7.2.1 depiler()

```
int depiler (  
    pile ** p )
```

Fonction permettant de depiler une valeur.

##### Parameters

<i>p</i>	- la pile
----------	-----------

##### Returns

valeur - la valeur qu'on vient de depiler

##### 4.7.2.2 empiler()

```
void empiler (
```

```
pile ** p,  
int valeur )
```

Fonction permettant d'empiler une valeur.

#### Parameters

<i>p</i>	- la pile
<i>valeur</i>	- la valeur a empiler

#### 4.7.2.3 estvide()

```
int estvide (  
    pile ** p )
```

Fonction permettant de verifier si une pile est vide.

#### Parameters

<i>p</i>	- la pile
----------	-----------

#### Returns

1 si la pile est vide  
0 si la pile n'est pas vide

#### 4.7.2.4 existe()

```
int existe (  
    pile * p,  
    int valeur )
```

Fonction permettant de verifier si une valeur existe dans la pile.

#### Parameters

<i>p</i>	- la pile
<i>valeur</i>	- la valeur dont on veut verifier la presence dans la pile

#### Returns

existe - 0 si la valeur n'existe pas dans la pile, 1 sinon

#### 4.7.2.5 vider()

```
void vider (  
    pile ** p )
```

Fonction permettant de vider la pile.

**Parameters**

<i>p</i>	- la pile
----------	-----------

**4.7.2.6 visualiser()**

```
void visualiser (
    pile * p )
```

Fonction permettant de visualiser la pile.

**Parameters**

<i>p</i>	- la pile
----------	-----------

**4.8 prediction.c File Reference**

Prediction.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "pile.h"
#include "prediction.h"
```

**Functions**

- char \* [conversionint](#) (int i)  
*Fonction de conversion d'un entier en chaine de caractere. Utile pour traiter les motifs de la base sequentielle.*
- int \* [conversionmotif](#) (char \*motif)  
*Fonction de conversion d'un motif en tableau d'entiers qu'il represente. Utile pour traiter les motifs de la base sequentielle.*
- char \* [conversiontmotif](#) (int \*tmotif, int taillemotif)  
*Fonction de conversion d'un tableau d'entiers en motif. Utile pour traiter les motifs de la base sequentielle.*
- int [score](#) (int contexte, char \*motif)  
*Fonction de calcul du score d'un motif dans un contexte. Le score correspond au nombre de motifs dans la base sequentielle qui correspondent avec le motif dont on veut calculer le score.*
- int [prediction](#) (int contexte, char \*motif)  
*Fonction de prediction pour un motif dans un contexte. Les predictions sont stockees dans un fichier prediction.txt.*

**4.8.1 Detailed Description**

Prediction.



**Author**

Stella Zevio

**Version**

1.0

**Date**

2015-06-20

Prediction medicale

## 4.8.2 Function Documentation

### 4.8.2.1 conversionint()

```
char * conversionint (
    int i )
```

Fonction de conversion d'un entier en chaine de caractere. Utile pour traiter les motifs de la base sequentielle.

**Parameters**

<i>i</i>	- l'entier a convertir
----------	------------------------

**Returns**

s - la chaine de caractere equivalente a l'entier

### 4.8.2.2 conversionmotif()

```
int conversionmotif (
    char * motif )
```

Fonction de conversion d'un motif en tableau d'entiers qu'il represente. Utile pour traiter les motifs de la base sequentielle.

**Parameters**

<i>motif</i>	- le motif qu'on souhaite convertir en tableau d'entiers
--------------	--

**Returns**

T - le tableau d'entiers represente par le motif

#### 4.8.2.3 conversiontmotif()

```
char * conversiontmotif (
    int * tmotif,
    int taillemotif )
```

Fonction de conversion d'un tableau d'entiers en motif. Utile pour traiter les motifs de la base sequentielle.

##### Parameters

<i>tmotif</i>	- le tableau d'entiers representant un motif
<i>taillemotif</i>	- la taille du tableau d'entiers

##### Returns

motif - le motif sous forme de chaine de caracteres

#### 4.8.2.4 prediction()

```
int prediction (
    int contexte,
    char * motif )
```

Fonction de prediction pour un motif dans un contexte. Les predictions sont stockees dans un fichier prediction.txt.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif sur lequel on veut realiser la prediction

##### Returns

motifscorrespondants - le nombre de motifs de la base qui contiennent l'integralite du motif du patient. Si le motif est 1 2 3 et que la base sequentielle contient le motif 1 4 2 3, il fera partie des motifs correspondants

#### 4.8.2.5 score()

```
int score (
    int contexte,
    char * motif )
```

Fonction de calcul du score d'un motif dans un contexte. Le score correspond au nombre de motifs dans la base sequentielle qui correspondent avec le motif dont on veut calculer le score.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif dont on calcule le score

## Returns

score - le score du motif dans le contexte

## 4.9 prediction.h File Reference

Prediction des suites d'un patient a partir de l'analyse des motifs de patients contenus dans une base sequentielle.

### Functions

- char \* [conversionint](#) (int i)  
*Fonction de conversion d'un entier en chaine de caractere. Utile pour traiter les motifs de la base sequentielle.*
- int \* [conversionmotif](#) (char \*motif)  
*Fonction de conversion d'un motif en tableau d'entiers qu'il represente. Utile pour traiter les motifs de la base sequentielle.*
- char \* [conversiontmotif](#) (int \*tmotif, int taillmotif)  
*Fonction de conversion d'un tableau d'entiers en motif. Utile pour traiter les motifs de la base sequentielle.*
- int [score](#) (int contexte, char \*motif)  
*Fonction de calcul du score d'un motif dans un contexte. Le score correspond au nombre de motifs dans la base sequentielle qui correspondent avec le motif dont on veut calculer le score.*
- int [prediction](#) (int contexte, char \*motif)  
*Fonction de prediction pour un motif dans un contexte. Les predictions sont stockees dans un fichier prediction.txt.*

### 4.9.1 Detailed Description

Prediction des suites d'un patient a partir de l'analyse des motifs de patients contenus dans une base sequentielle.

## Author

Stella Zevio

## Version

1.0

## Date

2015-06-20

Prediction medicale

### 4.9.2 Function Documentation

#### 4.9.2.1 [conversionint](#)()

```
char* conversionint (  
    int i )
```

Fonction de conversion d'un entier en chaine de caractere. Utile pour traiter les motifs de la base sequentielle.

**Parameters**

<i>i</i>	- l'entier a convertir
----------	------------------------

**Returns**

s - la chaine de caractere equivalente a l'entier

**4.9.2.2 conversionmotif()**

```
int* conversionmotif (
    char * motif )
```

Fonction de conversion d'un motif en tableau d'entiers qu'il represente. Utile pour traiter les motifs de la base sequentielle.

**Parameters**

<i>motif</i>	- le motif qu'on souhaite convertir en tableau d'entiers
--------------	--

**Returns**

T - le tableau d'entiers represente par le motif

**4.9.2.3 conversiontmotif()**

```
char* conversiontmotif (
    int * tmotif,
    int taillemotif )
```

Fonction de conversion d'un tableau d'entiers en motif. Utile pour traiter les motifs de la base sequentielle.

**Parameters**

<i>tmotif</i>	- le tableau d'entiers representant un motif
<i>taillemotif</i>	- la taille du tableau d'entiers

**Returns**

motif - le motif sous forme de chaine de caracteres

**4.9.2.4 prediction()**

```
int prediction (
    int contexte,
    char * motif )
```

Fonction de prediction pour un motif dans un contexte. Les predictions sont stockees dans un fichier prediction.txt.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif sur lequel on veut realiser la prediction

## Returns

motifscorrespondants - le nombre de motifs de la base qui contiennent l'integralite du motif du patient. Si le motif est 1 2 3 et que la base sequentielle contient le motif 1 4 2 3, il fera partie des motifs correspondants

## 4.9.2.5 score()

```
int score (
    int contexte,
    char * motif )
```

Fonction de calcul du score d'un motif dans un contexte. Le score correspond au nombre de motifs dans la base sequentielle qui correspondent avec le motif dont on veut calculer le score.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif dont on calcule le score

## Returns

score - le score du motif dans le contexte

## 4.10 predictionpartielle.c File Reference

Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "pile.h"
#include "prediction.h"
#include "predictionpartielle.h"
```

## Functions

- char \* [decoupermotif](#) (char \*motif, int debut, int fin)  
*Fonction permettant de scinder un motif.*
- int [scoremax](#) (int contexte, char \*motif\_1, char \*motif\_2, char \*motif\_3)  
*Fonction permettant de connaitre le score maximum de trois motifs. Utile lorsqu'on souhaite decouper un motif de trois hospitalisations en groupes de 2.*

- `pile * remplirListePredictions2` (int contexte, char \*motif)  
*Fonction permettant de remplir la liste des predictions pour un motif scinde en groupes de deux.*
- float `calculerProba2` (int contexte, int pred, char \*motif, char \*amotif)  
*Fonction permettant de calculer la probabilite d'une prediction pour un motif de deux hospitalisations.*
- `pile * remplirListePredictions1` (int contexte, char \*evenement\_1, char \*evenement\_2, char \*evenement\_3)  
*Fonction permettant de remplir la liste des predictions pour chaque hospitalisation prise de maniere isolee.*
- float `calculerProba1` (int contexte, int pred, char \*evenement\_1, char \*evenement\_2, char \*evenement\_3)  
*Fonction permettant de calculer la probabilite d'une prediction pour chaque hospitalisation prise de maniere isolee.*
- void `predictionpartielle` (int contexte, char \*motif)  
*Fonction permettant de realiser des predictions partielles sur un motif.*

#### 4.10.1 Detailed Description

Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle)

##### Author

Stella Zevio

##### Version

1.0

##### Date

2015-06-20

Prediction medicale

#### 4.10.2 Function Documentation

##### 4.10.2.1 calculerProba1()

```
calculerProba1 (
    int contexte,
    int pred,
    char * evenement_1,
    char * evenement_2,
    char * evenement_3 )
```

Fonction permettant de calculer la probabilite d'une prediction pour chaque hospitalisation prise de maniere isolee.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>pred</i>	- la prediction
<i>evenement_1</i>	- une hospitalisation
<i>evenement_2</i>	- une hospitalisation
<i>evenement_3</i>	- une hospitalisation

### Returns

proba - la probabilite de la prediction

#### 4.10.2.2 calculerProba2()

```
calculerProba2 (
    int contexte,
    int pred,
    char * motif,
    char * amotif )
```

Fonction permettant de calculer la probabilite d'une prediction pour un motif de deux hospitalisations.

### Parameters

<i>contexte</i>	- le contexte medical considere
<i>pred</i>	- la prediction
<i>motif</i>	- le motif de deux hospitalisations
<i>amotif</i>	- l'hospitalisation restante du patient

### Returns

proba - la probabilite de la prediction

#### 4.10.2.3 decoupermotif()

```
decoupermotif (
    char * motif,
    int debut,
    int fin )
```

Fonction permettant de scinder un motif.

### Parameters

<i>motif</i>	- l'entier a convertir
<i>debut</i>	- position de debut d'extraction de la sous chaine dans le motif
<i>fin</i>	- position de fin d'extraction de la sous chaine dans le motif

### Returns

nmotif - motif de la position debut a la position fin

#### 4.10.2.4 predictionpartielle()

```
predictionpartielle (
    int contexte,
    char * motif )
```

Fonction permettant de realiser des predictions partielles sur un motif.



## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif du patient

## 4.10.2.5 remplirListePredictions1()

```
remplirListePredictions1 (
    int contexte,
    char * evenement_1,
    char * evenement_2,
    char * evenement_3 )
```

Fonction permettant de remplir la liste des predictions pour chaque hospitalisation prise de maniere isolee.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>evenement</i> <sub>↔</sub> <i>_1</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_2</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_3</i>	- une hospitalisation

## Returns

listePredictions - la liste des predictions

## 4.10.2.6 remplirListePredictions2()

```
remplirListePredictions2 (
    int contexte,
    char * motif )
```

Fonction permettant de remplir la liste des predictions pour un motif scinde en groupes de deux.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif

## Returns

listePredictions - la liste des predictions

#### 4.10.2.7 scoremax()

```
scoremax (
    int contexte,
    char * motif_1,
    char * motif_2,
    char * motif_3 )
```

Fonction permettant de connaitre le score maximum de trois motifs. Utile lorsqu'on souhaite decouper un motif de trois hospitalisations en groupes de 2.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif_1</i>	- un motif
<i>motif_2</i>	- un motif
<i>motif_3</i>	- un motif

##### Returns

score\_max - le score maximum

## 4.11 predictionpartielle.h File Reference

Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle)

```
#include "pile.h"
```

### Functions

- char \* [decoupermotif](#) (char \*motif, int debut, int fin)  
*Fonction permettant de scinder un motif.*
- int [scoremax](#) (int contexte, char \*motif\_1, char \*motif\_2, char \*motif\_3)  
*Fonction permettant de connaitre le score maximum de trois motifs. Utile lorsqu'on souhaite decouper un motif de trois hospitalisations en groupes de 2.*
- pile \* [remplirListePredictions2](#) (int contexte, char \*motif)  
*Fonction permettant de remplir la liste des predictions pour un motif scinde en groupes de deux.*
- float [calculerProba2](#) (int contexte, int pred, char \*motif, char \*amotif)  
*Fonction permettant de calculer la probabilite d'une prediction pour un motif de deux hospitalisations.*
- pile \* [remplirListePredictions1](#) (int contexte, char \*evenement\_1, char \*evenement\_2, char \*evenement\_3)  
*Fonction permettant de remplir la liste des predictions pour chaque hospitalisation prise de maniere isolee.*
- float [calculerProba1](#) (int contexte, int pred, char \*evenement\_1, char \*evenement\_2, char \*evenement\_3)  
*Fonction permettant de calculer la probabilite d'une prediction pour chaque hospitalisation prise de maniere isolee.*
- void [predictionpartielle](#) (int contexte, char \*motif)  
*Fonction permettant de realiser des predictions partielles sur un motif.*

### 4.11.1 Detailed Description

Prediction partielle sur un motif de patient (cas ou il n'existe pas de correspondance exacte entre le motif du patient et la base sequentielle)

#### Author

Stella Zevio

#### Version

1.0

#### Date

2015-06-20

Prediction medicale

### 4.11.2 Function Documentation

#### 4.11.2.1 calculerProba1()

```
float calculerProba1 (
    int contexte,
    int pred,
    char * evenement_1,
    char * evenement_2,
    char * evenement_3 )
```

Fonction permettant de calculer la probabilite d'une prediction pour chaque hospitalisation prise de maniere isolee.

#### Parameters

<i>contexte</i>	- le contexte medical considere
<i>pred</i>	- la prediction
<i>evenement</i> <sub>↔</sub> <i>_1</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_2</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_3</i>	- une hospitalisation

#### Returns

proba - la probabilite de la prediction

#### 4.11.2.2 calculerProba2()

```
float calculerProba2 (
    int contexte,
    int pred,
    char * motif,
    char * amotif )
```

Fonction permettant de calculer la probabilité d'une prediction pour un motif de deux hospitalisations.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>pred</i>	- la prediction
<i>motif</i>	- le motif de deux hospitalisations
<i>amotif</i>	- l'hospitalisation restante du patient

##### Returns

proba - la probabilité de la prediction

#### 4.11.2.3 decoupermotif()

```
char* decoupermotif (
    char * motif,
    int debut,
    int fin )
```

Fonction permettant de scinder un motif.

##### Parameters

<i>motif</i>	- l'entier a convertir
<i>debut</i>	- position de debut d'extraction de la sous chaine dans le motif
<i>fin</i>	- position de fin d'extraction de la sous chaine dans le motif

##### Returns

nmotif - motif de la position debut a la position fin

#### 4.11.2.4 predictionpartielle()

```
void predictionpartielle (
    int contexte,
    char * motif )
```

Fonction permettant de realiser des predictions partielles sur un motif.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif du patient

## 4.11.2.5 remplirListePredictions1()

```
pile* remplirListePredictions1 (
    int contexte,
    char * evenement_1,
    char * evenement_2,
    char * evenement_3 )
```

Fonction permettant de remplir la liste des predictions pour chaque hospitalisation prise de maniere isolee.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>evenement</i> <sub>↔</sub> <i>_1</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_2</i>	- une hospitalisation
<i>evenement</i> <sub>↔</sub> <i>_3</i>	- une hospitalisation

## Returns

listePredictions - la liste des predictions

## 4.11.2.6 remplirListePredictions2()

```
pile* remplirListePredictions2 (
    int contexte,
    char * motif )
```

Fonction permettant de remplir la liste des predictions pour un motif scinde en groupes de deux.

## Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif</i>	- le motif

## Returns

listePredictions - la liste des predictions

#### 4.11.2.7 scoremax()

```
int scoremax (
    int contexte,
    char * motif_1,
    char * motif_2,
    char * motif_3 )
```

Fonction permettant de connaitre le score maximum de trois motifs. Utile lorsqu'on souhaite decouper un motif de trois hospitalisations en groupes de 2.

##### Parameters

<i>contexte</i>	- le contexte medical considere
<i>motif_1</i>	- un motif
<i>motif_2</i>	- un motif
<i>motif_3</i>	- un motif

##### Returns

score\_max - le score maximum

# Index

- calculerProba1
  - predictionpartielle.c, [24](#)
  - predictionpartielle.h, [29](#)
- calculerProba2
  - predictionpartielle.c, [25](#)
  - predictionpartielle.h, [29](#)
- contextemedical.c, [7](#)
  - conversionchar, [7](#)
- contextemedical.h, [8](#)
  - conversionchar, [8](#)
- conversionchar
  - contextemedical.c, [7](#)
  - contextemedical.h, [8](#)
- conversionint
  - prediction.c, [19](#)
  - prediction.h, [21](#)
- conversionmotif
  - prediction.c, [19](#)
  - prediction.h, [22](#)
- conversiontmotif
  - prediction.c, [19](#)
  - prediction.h, [22](#)
- creermotif
  - patient.c, [10](#)
  - patient.h, [12](#)
- decoupermotif
  - predictionpartielle.c, [25](#)
  - predictionpartielle.h, [30](#)
- depiler
  - pile.c, [13](#)
  - pile.h, [16](#)
- empiler
  - pile.c, [14](#)
  - pile.h, [16](#)
- estvide
  - pile.c, [14](#)
  - pile.h, [17](#)
- existe
  - pile.c, [14](#)
  - pile.h, [17](#)
- limitermotif
  - patient.c, [11](#)
  - patient.h, [12](#)
- main.c, [9](#)
- patient.c, [10](#)
  - creermotif, [10](#)
- limitermotif, [11](#)
- patient.h, [11](#)
  - creermotif, [12](#)
  - limitermotif, [12](#)
- Pile, [5](#)
- pile, [5](#)
- pile.c, [12](#)
  - depiler, [13](#)
  - empiler, [14](#)
  - estvide, [14](#)
  - existe, [14](#)
  - vider, [15](#)
  - visualiser, [15](#)
- pile.h, [15](#)
  - depiler, [16](#)
  - empiler, [16](#)
  - estvide, [17](#)
  - existe, [17](#)
  - vider, [17](#)
  - visualiser, [18](#)
- prediction
  - prediction.c, [20](#)
  - prediction.h, [22](#)
- prediction.c, [18](#)
  - conversionint, [19](#)
  - conversionmotif, [19](#)
  - conversiontmotif, [19](#)
  - prediction, [20](#)
  - score, [20](#)
- prediction.h, [21](#)
  - conversionint, [21](#)
  - conversionmotif, [22](#)
  - conversiontmotif, [22](#)
  - prediction, [22](#)
  - score, [23](#)
- predictionpartielle
  - predictionpartielle.c, [25](#)
  - predictionpartielle.h, [30](#)
- predictionpartielle.c, [23](#)
  - calculerProba1, [24](#)
  - calculerProba2, [25](#)
  - decoupermotif, [25](#)
  - predictionpartielle, [25](#)
  - remplirListePredictions1, [27](#)
  - remplirListePredictions2, [27](#)
  - scoremax, [27](#)
- predictionpartielle.h, [28](#)
  - calculerProba1, [29](#)
  - calculerProba2, [29](#)

- decoupermotif, [30](#)
- predictionpartielle, [30](#)
- remplirListePredictions1, [31](#)
- remplirListePredictions2, [31](#)
- scoremax, [31](#)
  
- remplirListePredictions1
  - predictionpartielle.c, [27](#)
  - predictionpartielle.h, [31](#)
- remplirListePredictions2
  - predictionpartielle.c, [27](#)
  - predictionpartielle.h, [31](#)
  
- score
  - prediction.c, [20](#)
  - prediction.h, [23](#)
- scoremax
  - predictionpartielle.c, [27](#)
  - predictionpartielle.h, [31](#)
  
- vider
  - pile.c, [15](#)
  - pile.h, [17](#)
- visualiser
  - pile.c, [15](#)
  - pile.h, [18](#)