

GRA

Généré par Doxygen 1.8.17

Chapter 1

gra

Gustatory Receptor family Assigner

Chapter 2

Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

MapGeneProfile

Structure permettant de stocker le header d'une séquence, le profil attribué et la E-value associée ??

Tokens

Structure permettant de stocker un tableau de chaînes de caractères ainsi que sa taille ??

Chapter 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

[gra.c](#)

Modifie un fichier multifasta pour retirer les gènes n'étant pas des gènes du goût, et ajoute dans le header des gènes restant la famille de gène correspondante ainsi que la E-value associée . ??

Chapter 4

Documentation des classes

4.1 Référence de la structure MapGeneProfile

Structure permettant de stocker le header d'une séquence, le profil attribué et la E-value associée.

Attributs publics

- char * **gene**
- char * **profile**
- char * **e_value**

4.1.1 Description détaillée

Structure permettant de stocker le header d'une séquence, le profil attribué et la E-value associée.

La documentation de cette structure a été générée à partir du fichier suivant :

- [gra.c](#)

4.2 Référence de la structure Tokens

Structure permettant de stocker un tableau de chaînes de caractères ainsi que sa taille.

Attributs publics

- size_t **size**
- char ** **tokens**

4.2.1 Description détaillée

Structure permettant de stocker un tableau de chaînes de caractères ainsi que sa taille.

La documentation de cette structure a été générée à partir du fichier suivant :

- [gra.c](#)

Chapter 5

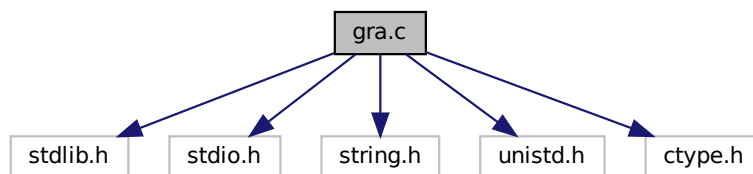
Documentation des fichiers

5.1 Référence du fichier gra.c

Modifie un fichier multifasta pour retirer les gènes n'étant pas des gènes du goût, et ajoute dans le header des gènes restant la famille de gène correspondante ainsi que la E-value associée.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>
```

Graphe des dépendances par inclusion de gra.c:



Classes

- struct [Tokens](#)

Structure permettant de stocker un tableau de chaînes de caractères ainsi que sa taille.

- struct [MapGeneProfile](#)

Structure permettant de stocker le header d'une séquence, le profil attribué et la E-value associée.

Fonctions

- **Tokens `split_string`** (const char *str, char *sep)
Divise une chaîne de caractère en plusieurs sous-chaîne selon un caractère séparateur.
- char **`complement`** (char c)
Renvoie le nucléotide complémentaire.
- char * **`reverse_complement`** (char *seq, int crlf)
Donne le complément inverse d'une séquence de nucléotide.
- const size_t **`count_gene`** (const char *filename)
Compte le nombre de séquences dans un fichier FASTA.
- FILE * **`hmmscan`** (const char *fasta, const char *profile, const char *e_value, const char *cpu)
Fonction appliquant la fonction hmmscan du logiciel HMMER.
- size_t **`parse_scan_res`** (FILE *scan_res, const size_t n_gene, MapGeneProfile *retrievedGenes)
Fonction de parsing du fichier table résultat de hmm_scan. La fonction va récupérer chaque référence de gène présent dans le fichier résultat ainsi que le profil associé
- char * **`get_filename`** (const char *filename)
Récupère le nom d'un fichier, en retirant le chemin et l'extension.
- char * **`default_output_name`** (const char *fasta)
Génère un nom par défaut pour le fichier de sortie.
- void **`fasta_output`** (const char *fasta, MapGeneProfile *retrievedGenes, size_t n_retrieved_genes, char *fa_out_name)
fonction écrivant dans un nouveau fichier fasta les gènes reconnus par les profils HMM, annoté avec le nom du profil l'ayant reconnu.
- const char ** **`get_parameters`** (int argc, char **argv, size_t *i)
Parse les arguments entrés en ligne de commande afin de récupérer les différents paramètres du programme, qui sont les suivants :
Usage: gra [-options] <seqfile>

 - h : list options.
 - p <hmddb> : scan <seqfile> against <hmddb>.
 - r : take the reverse complement of each sequence.
 - e <x> : set <x> as the E-value threshold. GRA will only report sequences with an E-value <= <x>.
 - Defaults is 0.9e-45.
 - o <f> : redirect output to the file <f>.
 - Default is "<seqfile>_GRA_OUTPUT.fasta".
 - If there are multiple fasta as input, default is used for each.
 - c <n> : Set the number of parallel worker threads to <n>. On multicore machines, the default is 2.
- char * **`write_rc_fasta`** (const char *fasta)
Écrit dans un nouveau fichier le complément inverse de chaque séquence du fichier fasta passé en paramètre.
- void **`run_gra`** (const char *fasta, const char *hmm, const char *e_value, char *fa_out, const char *cpu, const char *rc)
Lance une itération de GRA sur un fichier FASTA.
- int **`main`** (int argc, char **argv)

5.1.1 Description détaillée

Modifie un fichier multifasta pour retirer les gènes n'étant pas des gènes du goût, et ajoute dans le header des gènes restant la famille de gène correspondante ainsi que la E-value associée.

Auteur

Paul Bunel (paul.bunel@etu.umontpellier.fr)

Paramètres

<i>file</i>	Le fichier fasta d'entrée contenant les gènes à tester
-------------	--

Renvoie

Un nouveau fichier fasta contenant les gènes validés annotés

Version

0.1

Date

2022-06-30

5.1.2 Documentation des fonctions

5.1.2.1 `complement()`

```
char complement (
    char c )
```

Renvoie le nucléotide complémentaire.

Paramètres

<i>c</i>	un nucléotide
----------	---------------

Renvoie

char le nucléotide complémentaire de c.

5.1.2.2 `count_gene()`

```
const size_t count_gene (
    const char * filename )
```

Compte le nombre de séquences dans un fichier FASTA.

Paramètres

<i>filename</i>	nom du fichier FASTA
-----------------	----------------------

Renvoie

const size_t nombre de séquences

5.1.2.3 default_output_name()

```
char* default_output_name (
    const char * fasta )
```

Génère un nom par défaut pour le fichier de sortie.

Paramètres

<i>fasta</i>	nom du fichier FASTA d'entrée
--------------	-------------------------------

Renvoie

char* nom du fichier FASTA de sortie

5.1.2.4 fasta_output()

```
void fasta_output (
    const char * fasta,
    MapGeneProfile * retrievedGenes,
    size_t n_retrieved_genes,
    char * fa_out_name )
```

fonction écrivant dans un nouveau fichier fasta les gènes reconnus par les profils HMM, annoté avec le nom du profil l'ayant reconnu.

Paramètres

<i>fasta</i>	le fichier fasta d'entrée contenant tous les gènes à tester
<i>retrievedGenes</i>	tableau de MapGeneProfile contenant les gènes reconnus
<i>n_retrieved_genes</i>	le nombre de gènes reconnus
<i>fa_out_name</i>	le nom du fichier FASTA de sortie

5.1.2.5 get_filename()

```
char* get_filename (
    const char * filename )
```

Récupère le nom d'un fichier, en retirant le chemin et l'extension.

Paramètres

<i>filename</i>	le chemin vers un fichier
-----------------	---------------------------

Renvoie

char* le nom tronqué du fichier

5.1.2.6 get_parameters()

```
const char** get_parameters (
    int argc,
    char ** argv,
    size_t * j )
```

Parse les arguments entrés en ligne de commande afin de récupérer les différents paramètres du programme, qui sont les suivants :

Usage: gra [-options] <seqfile>

-h : list options.

-p <hmmdb> : scan <seqfile> against <hmmdb>.

-r : take the reverse complement of each sequence.

-e <x> : set <x> as the E-value threshold. GRA will only report sequences with an E-value <= <x>. Defaults is 0.9e-45.

-o <f> : redirect output to the file <f>.

Default is "<seqfile>_GRA_OUTPUT.fasta".

If there are multiple fasta as input, default is used for each.

-c <n> : Set the number of parallel worker threads to <n>. On multicore machines, the default is 2.

Paramètres

<i>argc</i>	nombre d'argument
<i>argv</i>	les arguments
<i>j</i>	pointeur vers un size_t qui sera modifié

Renvoie

char** Un tableau de chaînes de caractère contenant les valeurs de chaque paramètre

5.1.2.7 hmmscan()

```
FILE* hmmscan (
    const char * fasta,
    const char * profile,
    const char * e_value,
    const char * cpu )
```

Fonction appliquant la fonction *hmmscan* du logiciel HMMER.

Paramètres

<i>fasta</i>	le fichier FASTA que l'on passe en paramètre de HMMER
<i>profile</i>	le profil HMM à utiliser
<i>e_value</i>	la E-value minimale à afficher
<i>cpu</i>	le nombre de threads que l'on veut utiliser

Renvoie

FILE* Un flux vers la sortie de HMMER.

5.1.2.8 `parse_scan_res()`

```
size_t parse_scan_res (
    FILE * scan_res,
    const size_t n_gene,
    MapGeneProfile * retrievedGenes )
```

Fonction de parsing du fichier table résultat de `hmm_scan`. La fonction va récupérer chaque référence de gène présent dans le fichier résultat ainsi que le profil associé

Paramètres

<i>scan_res</i>	le flux vers la sortie de HMMER
<i>n_gene</i>	nombre de gène dans le fichier FASTA en entrée du programme
<i>retrievedGenes</i>	tableau de MapGeneProfile qui sera modifié dans la fonction

Renvoie

size_t le nombre de gènes reconnus par HMMER

5.1.2.9 `reverse_complement()`

```
char* reverse_complement (
    char * seq,
    int crlf )
```

Donne le complément inverse d'une séquence de nucléotide.

Paramètres

<i>seq</i>	une séquence de nucléotide
<i>crlf</i>	type de fichier

Renvoie

char* Le complément inverse de seq.

5.1.2.10 run_gra()

```
void run_gra (
    const char * fasta,
    const char * hmm,
    const char * e_value,
    char * fa_out,
    const char * cpu,
    const char * rc )
```

Lance une itération de GRA sur un fichier FASTA.

Paramètres

<i>fasta</i>	Le nom du fichier FASTA en entrée
<i>hmm</i>	Le nom du profil HMM à utiliser
<i>e_value</i>	La E-value minimale à afficher
<i>fa_out</i>	Le nom du fichier FASTA en sortie
<i>cpu</i>	Le nombre de thread que l'on veut utiliser
<i>rc</i>	0 = on récupère le fichier fasta, 1 = on prend le complément inverse de chaque séquence

5.1.2.11 split_string()

```
Tokens split_string (
    const char * str,
    char * sep )
```

Divise une chaîne de caractère en plusieurs sous-chaîne selon un caractère séparateur.

Paramètres

<i>str</i>	la chaîne de caractère que l'on veut diviser
<i>sep</i>	le séparateur

Renvoie

Tokens un élément de la structure **Tokens** contenant toutes les divisions de la chaîne str.

5.1.2.12 write_rc_fasta()

```
char* write_rc_fasta (
    const char * fasta )
```

Ecrit dans un nouveau fichier le complément inverse de chaque séquence du fichier fasta passé en paramètre.

Paramètres

<i>fasta</i>	Un fichier FASTA
--------------	------------------

Renvoie

char* Le nom du fichier dans lequel il a écrit