Alpha Team

Generato da Doxygen 1.13.2

Capitolo 1

Indice dei namespace

1.1 Lista dei namespace

Questa è l'elenco di tutti i namespace con una loro breve descrizione:

Capitolo 2

Indice dei tipi composti

2.1 Elenco dei tipi composti

Queste sono le classi, le struct, le union e le interfacce con una loro breve descrizione:

Capitolo 3

Indice dei file

3.1 Elenco dei file

Questo è un elenco di tutti i file con una loro breve descrizione:

Capitolo 4

Documentazione dei namespace

4.1 Riferimenti per il namespace AlphaTeam

Namespace

- namespace Blast
- namespace Sequence

4.2 Riferimenti per il namespace AlphaTeam.Blast

Funzioni

- pd.DataFrame create_query_df (list kmer_query_list, str nome_colonna)
- pd.DataFrame create_sub_df (list kmer_sub_list)
- pd.DataFrame fill_sub_df (pd.DataFrame df, list kmer_subject_list)
- pd.DataFrame create_df_with_positions (pd.DataFrame query_df, pd.DataFrame subject_df, str filename)
- pd.DataFrame find_seeds (pd.DataFrame df, str filename, int kmer_length=22)
- str or None get_sequence (str header, list list_partenza_subject)
- int calculate_score (str sequence_1, str sequence_2)
- tuple[str, str] handle_gaps (str finestra_aggiunta_gap_query, str finestra_aggiunta_gap_sub, str finestra_↩ mismatch_query, str finestra_mismatch_sub, int gap_size, str gap_target, int x_max)
- tuple[int, int] find_mismatch_window (str sequence_query_ext, str sequence_sub_ext, int x_max)
- tuple[str, str] extend_seed_right (str sequence_query_ext, str sequence_sub_ext, int gap_size, bool perform_gap, str gap_target, int x_max)
- tuple[list, list, list, list] extend_seed (pd.DataFrame df_seeds, tuple query_partenza, list list_partenza_subject, int x_max)
- pd.DataFrame create_results_table (list cont_hsp_query, list cont_hsp_sub, list cont_hsp_score, list col, str filename)
- tuple[list, list, list] metrics_for_blast (pd.DataFrame best_alignment_df, tuple query_partenza, list list_↩ partenza_subject)
- pd.DataFrame blast_result_df (pd.DataFrame best_alignment_query, tuple query_partenza, list list_↩ partenza_subject, str filename)
- None print_alignment (pd.DataFrame best_alignment_df, str output_file, pd.DataFrame result_df)

Variabili

- dict transizione = {'A': 'G', 'G': 'A', 'C': 'T', 'T': 'C'}
- dict trasversione

4.2.1 Documentazione delle funzioni

4.2.1.1 blast_result_df()

```
pd.DataFrame AlphaTeam.Blast.blast_result_df (
             pd.DataFrame best_alignment_query,
             tuple query_partenza,
             list list_partenza_subject,
             str filename)
```

```
La funzione migliora un dataframe di allineamento BLAST con le metriche calcolate e le salva in un DataFrame c
Parameters
----------
best_alignment_query (pd.DataFrame): dataframe contenente i migliori allineamenti.
query_partenza (tuple): tupla in cui il secondo elemento della query è la sequenza query usata per le analisi
list_partenza_subject (list): lista di sequenze subject che ritorna la lunghezza delle subject per il calcolo
filename (str): nome del file nel quale il dataframe verrà salvato.
Return
------
best_alignment_query (pd.DataFrame): dataframe salvato in un file .csv con lo specifico filename.
```

4.2.1.2 calculate_score()

```
int AlphaTeam.Blast.calculate_score (
             str sequence_1,
             str sequence_2)
```

```
La funzione calcola il punteggio di somiglianza tra due sequenze in base a regole specifiche di corrispondenza
Parameters
----------
 sequence_1 (str): prima sequenza da confrontare.
 sequence_2 (str): seconda sequenza da confrontare.
Return
------
 score (int): punteggio di somiglianza.
```

4.2.1.3 create_df_with_positions()

```
pd.DataFrame AlphaTeam.Blast.create_df_with_positions (
             pd.DataFrame query_df,
             pd.DataFrame subject_df,
             str filename)
```

```
La funzione combina i due dataframe in input, tramite inner join, per generarne uno nuovo contenente posizioni
di k-mer tra i due. Salva poi il risultato in un file CSV e restituisce il dataframe finale.
Parameters
----------
 query_df (pd.DataFrame): dataframe con i k-mers realtivi alla sequenza query e relative posizioni.
 subject_df (pd.DataFrame): dataframe con i k-mers realtivi alle sequenze subject e relative posizioni nella l
 filename (str): nome del file (senza estensione) in cui il dataframe risultante verrà salvato in formato .csv
Return
------
 final_df (pd.DataFrame): dataframe risultante che mappa i k-mer con le posizioni della query e dei subject in
   salvato anche come file CSV.
```

4.2.1.4 create_query_df()

```
pd.DataFrame AlphaTeam.Blast.create_query_df (
             list kmer_query_list,
             str nome_colonna)


La funzione prende una lista di input (kmer_query_list) e costruisce un dataframe Pandas
organizzato in modo specifico con intestazioni e indici derivati dalla lista. Inoltre riempie il df creato con
Parameters
----------
 kmer_query_list (list): Lista alternata di intestazioni (index pari) e liste di k-mers (index dispari).
 nome_colonna (str): Nome di colonna di intestazione da ricercare nel df, per capirne la lunghezza
Return
------
 df (pd.DataFrame) : dataframe con relativi valori.
```

4.2.1.5 create_results_table()

```
pd.DataFrame AlphaTeam.Blast.create_results_table (
             list cont_hsp_query,
             list cont_hsp_sub,
             list cont_hsp_score,
             list col,
             str filename)


Crea una result table dagli HSP (High-scoring Segment Pair) data e li salva in un file .csv.
La tabella include query sequence, subject sequences, headers e gli scores, ordinati in modo decrescente.
Parameters
----------
cont_hsp_query (list): lista delle sequenze query derivate dall'analisi dell'HSP.
cont_hsp_sub (list): lista delle sequenze subject derivate dall'analisi dell'HSP.
cont_hsp_score (list): lista di score corrispondente agli HSP.
col (list): elenco di headers o identificatori per i risultati HSP.
filename (str):nome del file (senza estensione) nel quale il result-table verrà salvato.
Return
------
results_df (pd.DataFrame): salvato come file .csv.Un pandas DataFrame contenente
  i risultati degli HSP con le colonne 'hsp_query','hsp_sub','header','hsp_score'.
```

4.2.1.6 create_sub_df()

```
pd.DataFrame AlphaTeam.Blast.create_sub_df (
             list kmer_sub_list)


La funzione  costruisce un DataFrame Pandas organizzato in modo specifico con intestazioni e indici derivati d
In particolare, gli indici sono unici e disposti in ordine alfabetico.
Parameters
----------
kmer_sub_list (list): lista alternata di intestazioni delle sequenze e liste di k-mers relativi ad ogni sequen
Return
------
df (pd.DataFrame): dataframe che ha come indici i k-mers unici e in ordine alfabetico e, come nomi delle colon
```

4.2.1.7 extend_seed()

```
tuple[list,list,list,list] AlphaTeam.Blast.extend_seed (
            pd.DataFrame df_seeds,
            tuple query_partenza,
            list list_partenza_subject,
            int x_max)

Estende gli allineamenti dei seed iterando su di essi in un DataFrame fra la sequenza query e le molteplici se
Gestisce i mismatches, i gaps, e gli allineamenti continui laddove possibile.
Parameters
----------
df_seeds (pd.DataFrame): dataframe contenente i seeds per ogni colonna delle sequenze dei subject.
query_partenza (tuple): tupla contenente l'header della query e la sua sequenza.
list_partenza_subject (list): lista di tuple, in cui ogni tupla contiene l'header della sequenza subject e la
x_max (int): numero massimo di mismatches consecutivi consentito prima dell'arresto dell'estensione dell'allin
Return
------
tuple[list,list,list,list]: tupla composta da:
-contenitore_hsp_query (list): lista degli allineamenti della query.
-contenitore_hsp_sub (list): lista degli allineamenti delle subject estese.
-contenitore_score (list): lista degli scores per ogni allineamento esteso, in cui lo score si basa sulla lunç
-contenitore_ref (list): lista degli header dei subject del corrispettivo allineamento esteso.
```

4.2.1.8 extend_seed_right()

```
tuple[str,str] AlphaTeam.Blast.extend_seed_right (
            str sequence_query_ext,
            str sequence_sub_ext,
            int gap_size,
            bool perform_gap,
            str gap_target,
            int x_max)

Estende l'allineamento della query e delle sequenze subject da destra, inserisce i gap quando è necessario.
Parameters
----------
sequence_query_ext (str): sequenza query da estendere.
sequence_sub_ext (str): sequenza subject da estendere.
gap_size (int): numero massimo di gaps consentiti durante l'estensione.
perform_gap (bool): se True, i gaps sono inseriti per ottimizzare l'allineamento quando i mismatches superano
gap_target (str):specifica dove introdurre i gaps: "query" aggiunge i gaps alla sequenza query e "subject" li
x_max (int): threshold per i mismatches consecutivi necessari per attivare l'inserimento dei gaps o fermare l'
Return
------
tuple[str,str]: tupla composta da:
-extension_right_query (str): sequenza query estesa.
-extension_right_sub (str): sequenze subject estese.
```

4.2.1.9 fill_sub_df()

```
pd.DataFrame AlphaTeam.Blast.fill_sub_df (
            pd.DataFrame df,
            list kmer_subject_list)

La funzione prende in input un dataframe pre-esistente e ne riempie le celle ricavando informazioni dalla list
In particolare, essa verifica se ogni K-mer è presente nella lista dei k-mers associata ad ogni colonna e inse
l'indice che il k-mer occupa nella lista se esso è presente in quest'ultima.
Parameters
----------
 df (pd.DataFrame): dataframe iniziale
 kmer_subject_list (list): lista alternata di intestazioni delle sequenze e liste di k-mers relativi ad ogni s
Return
------
 df (pd.DataFrame): dataframe completo di valori.
```

4.2.1.10 find_mismatch_window()

```
tuple[int,int] AlphaTeam.Blast.find_mismatch_window (
            str sequence_query_ext,
            str sequence_sub_ext,
            int x_max)
```

```
La funzione confronta le basi nelle stesse posizioni delle sequenze in input assegnando un punteggio positivo
Ogni mismatch consecutivo incrementa un contatore. Se il numero di mismatch consecutivi raggiunge il valore sc
il ciclo si interrompe.
Parameters
----------
 sequence_query_ext (str): sequenza estesa da query da confrontare.
 sequence_sub_ext (str): sequenza estesa da subject da confrontare.
 x_max (int): numero massimo di mismatch consecutivi consentiti prima di interrompere il confronto.
Return
------
tuple[int,int]: tupla composta da:
 -last_valid_index (int): indice dell'ultima posizione valida prima dei mismatch consecutivi
 -mismatch_consecutivi (int): numero di mismatch consecutivi rilevati.
```

4.2.1.11 find_seeds()

```
pd.DataFrame AlphaTeam.Blast.find_seeds (
            pd.DataFrame df,
            str filename,
            int  kmer_length = 22)
```

```
La funzione identifica regioni di interesse ("seeds") basandosi sulle tuple (query_start, subject_start) e sul
Processa un dataframe contenente tuple rappresentanti posizioni iniziali di k-mer per creare un nuovo datafram
che includono le posizioni iniziali e finali. I risultati vengono salvati in un file CSV.
Parameters
----------
 df (pd.DataFrame): dataframe con informazioni realtive alle posizioni dei k-mers in seqeunze query e subject.
 filename (str): nome del file (senza estensione) in cui il dataframe risultante verrà salvato in formato .csv
 kmer_length (int): lunghezza dei k-mers.
Return
------
 new_df (pd.DataFrame): dataframe con i seeds organizzati, dove ogni seed include posizioni iniziali e finali.
    I risultati sono salvati anche nel file CSV specificato.
```

4.2.1.12 get_sequence()

```
str or None AlphaTeam.Blast.get_sequence (
            str header,
            list list_partenza_subject)
```

```
Recupera la sequenza corrispondente a un dato header.
Parameters
----------
 header (str): header del subject.
 list_partenza_subject (list): lista di tuple, ogni tupla composta da header e sequenza associata.
Return
------
str o None
-item[1] (str): se trova corrispondenza
-None se non trova niente.
```

4.2.1.13 handle_gaps()

```
tuple[str, str] AlphaTeam.Blast.handle_gaps (
             str finestra_aggiunta_gap_query,
             str finestra_aggiunta_gap_sub,
             str finestra_mismatch_query,
             str finestra_mismatch_sub,
             int gap_size,
             str gap_target,
             int x_max)

La funzione gestisce i gap aggiungendoli nella sequenza individuata da gap_target.
Parameters
----------
 finestra_aggiunta_gap_query (str):  sequenza della query dove aggiungere i gap.
 finestra_aggiunta_gap_sub (str): sequenza della subject dove aggiungere i gap.
 finestra_mismatch_query (str): finestra della query che presenta mismatch.
 finestra_mismatch_sub (str): finestra della subject che presenta mismatch.
 gap_size (int): numero massimo di gap consentiti durante l'estensione
 gap_target (str): specifica dove aggiungere i gap. Può essere 'query' o 'subject'.
 x_max (int) : valore massimo negativo del punteggio accettabile per le sequenze.
Return
------
tuple[str,str]: tupla composta da:
 -finestra_aggiunta_gap_query (str): finestra di sequenza query con aggiunta dei gap, o '*' se i gap non migli
 -finestra_aggiunta_gap_sub (str): finestra di sequenza subject con aggiunta dei gap, o '*' se i gap non migli
```

4.2.1.14 metrics_for_blast()

```
tuple[list,list,list] AlphaTeam.Blast.metrics_for_blast (
             pd.DataFrame best_alignment_df,
             tuple query_partenza,
             list list_partenza_subject)

La funzione calcola le metriche degli allineamenti del BLAST, includendo query_coverage, E-value e la percentu
Parameters
----------
best_alignment_df (pd.Dataframe): dataframe contenente gli allineamenti migliori.
query_partenza (tuple): tupla in cui il secondo elemento è una sequenza query usata per le analisi del BLAST.
list_partenza_subject (list): lista di sequenze subject che recupera la lunghezza dei subject per il calcolo d
Return
------
tuple[list,list,list]: tupla composta da:
-query_cov_list (list): lista della copertura in percentuale della query per ogni allineamento.
-e_value_list (list): lista degli E-values calcolato per ogni allineamento.
-identity_list (list): lista dei valori in percentuale dell'identità per ogni allineamento.
```

4.2.1.15 print_alignment()

```
None AlphaTeam.Blast.print_alignment (
             pd.DataFrame best_alignment_df,
             str output_file,
             pd.DataFrame result_df)

La funzione rappresenta i risultati degli allineamenti in un file, includendo l'allineamento di sequenza, le m
Parameters
----------
best_alignment_df (pd.DataFrame): dataframe contenente i migliori allineamenti in cui nelle colonne inseriamo
output_file (str): percorso del file di output in cui verranno scritti i risultati dell'allineamento formattat
result_df (pd.DataFrame): dataframe contenente le metriche calcolate per ogni allineamento
Return
------
None
    Scrive i dettagli dell'allineamento nello specifico output finale.
```

4.2.2 Documentazione delle variabili

4.2.2.1 transizione

```
dict AlphaTeam.Blast.transizione = {'A': 'G', 'G': 'A', 'C': 'T', 'T': 'C'}
```

4.2.2.2 trasversione

```
dict AlphaTeam.Blast.trasversione
```

Valore iniziale:

```
00001 =  {
00002     'A': ['C', 'T'],
00003     'C': ['A', 'G'],
00004     'G': ['C', 'T'],
00005     'T': ['A', 'G']
00006 }
```

4.3 Riferimenti per il namespace AlphaTeam.Sequence

Composti

- class Sequence

Capitolo 5

Documentazione delle classi

5.1 Riferimenti per la classe AlphaTeam.Sequence.Sequence

Membri pubblici

- __init__ (self, sequence_file)
- parse_file (self, num_sequences=None)
- list kmer_indexing (self, int k)
- list kmer_indexing_comp_rev (self, int k)
- slice_for_two_query (self)

Attributi pubblici

- query_partenza = None
- query_partenza_2 = None
- kmer_query1 = None
- kmer_query2 = None
- kmer_query1r = None
- kmer_query2r = None
- sequence_file = sequence_file
- seq_list = None
- forward_kmers = None
- comp_rev_kmers = None

5.1.1 Descrizione dettagliata

```
Una classe che gestisce e processa i file di sequenza e
indicizza i k-mer per le sequenze forward e le loro complementari revertite.
Attributes
----------
sequence_file: str
    Il percorso del file della sequenza di input (.fasta, .fa, or compressed .gz).

seq_list: list
    Parsa le sequenze come liste di tuple. Di default è None.

forward_kmers: list
    Sequenze indicizzate di k-mer del filamento forward. Di default è None.
```

```
comp_rev_kmers: list
    Sequenze indicizzate di k-mer del filamento complementare revertito. Di default è None.

query_partenza e query_partenza_2 : list
    Liste di query generate dalla funzione parse file. Di default impostate a None.

kmer_query1 : list
    Lista di kmer contenuti nella Query1. Di default impostata a None.

kmer_query2 : list
    Lista di kmer contenuti nella Query2. Di default impostata a None.

kmer_query1r : list
    Lista di kmer contenuti nella Query1 complementare revertita. Di default impostata a None.

kmer_query2r : list
    Lista di kmer contenuti nella Query2 complementare revertita. Di default impostata a None.

Methods
-------
    parse_file():
        Ritorna un file parsato.
    kmer_indexing(k: int) -> list:
        Genera i kmer indicizati per le sequenze forward.
    kmer_indexing_comp_rev(k: int) -> list:
        Genera i kmer indicizati per le sequenze complementari reverite.
```

5.1.2 Documentazione dei costruttori e dei distruttori

5.1.2.1 __init__()

```
AlphaTeam.Sequence.Sequence.__init__ (
              self,
              sequence_file)

Costruisce tutti gli attributi necessari per l'oggetto Sequence.
Parameters
----------
sequence_file: str
    Il percorso (path) del file della sequenza di input (.fasta, .fa, or compressed .gz).
seq_list: list
    Parsa le sequenze come lista di tuple (ID, sequence). Di default è None.
forward_kmers: list
    Sequenze indicizzate di k-mer del filamento forward. Di default è None.
comp_rev_kmers: list
    Sequenze indicizzate di k-mer del filamento complementare revertito. Di default è None.
```

5.1.3 Documentazione delle funzioni membro

5.1.3.1 kmer_indexing()

```
list AlphaTeam.Sequence.Sequence.kmer_indexing (
              self,
             int k)

Genera indici k-mer per le sequenze forward.
Args
----
k: int
    la lunghezza dei k-mers.
Return
------
complete_list: list
    Una lista contenente gli ID e le loro sequenze k-mer corrispondenti.
```

5.1.3.2 kmer_indexing_comp_rev()

```
list AlphaTeam.Sequence.Sequence.kmer_indexing_comp_rev (
              self,
             int k)
```

```
Genera indici k-mer per le sequenze complementari revertite.
Args
----
k: int
    La lunghezza dei k-mers.
Return
------
complete_list_comp_rev: list
    Una lista contenente gli ID e le loro sequenze k-mer complementari revertite corrispondenti.
```

5.1.3.3 parse_file()

```
AlphaTeam.Sequence.Sequence.parse_file (
              self,
              num_sequences = None)
```

```
Parsa il file sequenza in formato FASTA o nei formati compressi.

Parameters
----------
num_sequences: int
    Numero di sequenze da processare. Di default impostato a None.
Return
------
seq_list: list
    Una lista di sequenze parsate[(ID, sequence)]
```

5.1.3.4 slice_for_two_query()

```
AlphaTeam.Sequence.Sequence.slice_for_two_query (
              self)
```

5.1.4 Documentazione dei membri dato

5.1.4.1 comp_rev_kmers

```
AlphaTeam.Sequence.Sequence.comp_rev_kmers = None
```

5.1.4.2 forward_kmers

```
AlphaTeam.Sequence.Sequence.forward_kmers = None
```

5.1.4.3 kmer_query1

```
AlphaTeam.Sequence.Sequence.kmer_query1 = None
```

5.1.4.4 kmer_query1r

```
AlphaTeam.Sequence.Sequence.kmer_query1r = None
```

5.1.4.5 kmer_query2

```
AlphaTeam.Sequence.Sequence.kmer_query2 = None
```

5.1.4.6 kmer_query2r

```
AlphaTeam.Sequence.Sequence.kmer_query2r = None
```

5.1.4.7 query_partenza

```
AlphaTeam.Sequence.Sequence.query_partenza = None
```

5.1.4.8 query_partenza_2

```
AlphaTeam.Sequence.Sequence.query_partenza_2 = None
```

5.1.4.9 seq_list

```
AlphaTeam.Sequence.Sequence.seq_list = None
```

5.1.4.10 sequence_file

```
AlphaTeam.Sequence.Sequence.sequence_file = sequence_file
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- Sequence.py

Capitolo 6

Documentazione dei file

6.1 Riferimenti per il file Blast.py

Namespace

- namespace AlphaTeam
- namespace AlphaTeam.Blast

Funzioni

- pd.DataFrame AlphaTeam.Blast.create_query_df (list kmer_query_list, str nome_colonna)
- pd.DataFrame AlphaTeam.Blast.create_sub_df (list kmer_sub_list)
- pd.DataFrame AlphaTeam.Blast.fill_sub_df (pd.DataFrame df, list kmer_subject_list)
- pd.DataFrame AlphaTeam.Blast.create_df_with_positions (pd.DataFrame query_df, pd.DataFrame subject↩ _df, str filename)
- pd.DataFrame AlphaTeam.Blast.find_seeds (pd.DataFrame df, str filename, int kmer_length=22)
- str or None AlphaTeam.Blast.get_sequence (str header, list list_partenza_subject)
- int AlphaTeam.Blast.calculate_score (str sequence_1, str sequence_2)
- tuple[str, str] AlphaTeam.Blast.handle_gaps (str finestra_aggiunta_gap_query, str finestra_aggiunta_gap_↩ sub, str finestra_mismatch_query, str finestra_mismatch_sub, int gap_size, str gap_target, int x_max)
- tuple[int, int] AlphaTeam.Blast.find_mismatch_window (str sequence_query_ext, str sequence_sub_ext, int x_max)
- tuple[str, str] AlphaTeam.Blast.extend_seed_right (str sequence_query_ext, str sequence_sub_ext, int gap↩ _size, bool perform_gap, str gap_target, int x_max)
- tuple[list, list, list, list] AlphaTeam.Blast.extend_seed (pd.DataFrame df_seeds, tuple query_partenza, list list_partenza_subject, int x_max)
- pd.DataFrame AlphaTeam.Blast.create_results_table (list cont_hsp_query, list cont_hsp_sub, list cont_hsp↩ _score, list col, str filename)
- tuple[list, list, list] AlphaTeam.Blast.metrics_for_blast (pd.DataFrame best_alignment_df, tuple query_↩ partenza, list list_partenza_subject)
- pd.DataFrame AlphaTeam.Blast.blast_result_df (pd.DataFrame best_alignment_query, tuple query_↩ partenza, list list_partenza_subject, str filename)
- None AlphaTeam.Blast.print_alignment (pd.DataFrame best_alignment_df, str output_file, pd.DataFrame result_df)

Variabili

- dict AlphaTeam.Blast.transizione = {'A': 'G', 'G': 'A', 'C': 'T', 'T': 'C'}
- dict AlphaTeam.Blast.trasversione

6.2 Riferimenti per il file Sequence.py

Composti

- class AlphaTeam.Sequence.Sequence

Namespace

- namespace AlphaTeam
- namespace AlphaTeam.Sequence