SARST2 Manuel de l'utilisateur

Version française

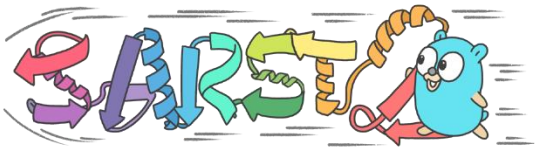


Table des matières

1. À propos de ce logiciel

SARST2 (Structural Similarity Search Aided by Ramachandran Sequential Transformation, version 2) est un algorithme d'alignement de structures protéiques de haute performance. Il prend en charge à la fois les recherches de similarité structurelle dans une base de données à l'aide d'une protéine de requête donnée, ainsi que les alignements de structures par paires entre deux structures protéiques.

Ce logiciel est publié conjointement avec l'article suivant et sera fréquemment mis à jour via les URL fournies dans la publication :

Titre	SARST2, a high-throughput protein structure alignment algorithm for searching massive databases
Auteurs	Wei-Cheng Lo*, Arieh Warshel, Chia-Hua Lo, Chia Yee Choke, Yan-Jie Li, Shih-Chung Yen, Jyun-Yi Yang and Shih-Wen Weng
Institut	Institute of Bioinformatics and Systems Biology, National Yang Ming Chiao Tung University, Hsinchu, Taiwan, Republic of China

*Auteur correspondant (WadeLo@nycu.edu.tw)

URLs de téléchargement:
https://github.com/NYCU-10lab/sarst
https://10lab.ceb.nycu.edu.tw/sarst2

2. Téléchargement, décompression et installation

La dernière version du programme SARST2 et des bases de données cibles préformatées sont disponibles aux URL listées ci-dessus.

Après avoir téléchargé une archive compressée, utilisez les utilitaires tar, gzip ou zip pour extraire les fichiers, selon le format de l'archive. Une fois extraits, les fichiers du programme SARST2 sont fournis sous forme de binaires exécutables précompilés et ne nécessitent aucune installation.

Par exemple, si vous téléchargez l'archive SARST2-v2.0.30-Linux.x86_64.tar.gz, vous pouvez la décompresser sous Linux à l'aide de la commande suivante :

```
tar xfp SARST2-v2.0.30-Linux.x86_64.tar.gz
```

Après l'extraction, vous pouvez exécuter le programme sarst2 avec les commandes suivantes :

```
cd SARST2-v2.0.30-Linux.x86_64/bin
chmod +x sarst2
./sarst2 -h
```

3. Contenu des dossiers du logiciel

bin	Exécutables 64 bits pour Linux, Windows et macOS.
dat	Exemples de fichiers de test PDB et SCOP.
doc	Manuels d'utilisation en plusieurs langues.

4. Guide rapide

Ce progiciel contient trois programmes principaux, décrits ci-dessous :

sarst2	Implémentation de l'algorithme d'alignement de structures protéiques SARST2.
formatdb	Un outil de formatage de base de données qui permet aux utilisateurs de créer des bases de données cibles personnalisées pour la recherche et l'alignement de structures.
readdb	Un outil pour extraire les séquences d'acides aminés protéiques et les séquences de structures encodées linéairement stockées dans une base de données cible préformatée.

L'exécution de l'un des programmes ci-dessus sans paramètres (ou avec -h) affichera un bref message d'aide textuel pour ce programme.

Linux, macOS

```
./sarst2
./formatdb
./readdb
./sarst2 -h
./formatdb -h
./readdb -h
```

Windows (Cmd ou PowerShell)

```
.\sarst2.exe
.\formatdb.exe
.\readdb.exe
.\sarst2.exe -h
.\formatdb.exe -h
.\readdb.exe -h
```

5. Manuel : sarst2

5.1 Utilisation

```
./sarst2    structure requête   structure(s) sujet      [Options]
            ----------------    -----------------
            > un fichier PDB/CIF > peut être
                                 1. -db + une base de données préformatée
                                 2. une liste de fichiers PDB/CIF
                                 3. des dossiers de fichiers PDB/CIF
                                 4. un fichier PDB/CIF (par paire)
```

5.2 Options du programme

-db	[str]	La base de données cible des structures sujets à rechercher. (par défaut : aucun)
-brief	[int]	Nombre de structures sujets pour afficher des résumés sur une ligne. (par défaut : 500)
-detail	[int]	Nombre de structures sujets pour afficher des données d'alignement détaillées. (par défaut : 500)
-t	[int]	Nombre de threads (fils d'exécution). Il doit être ≥ 0 ; avec 0, tous les processeurs seront utilisés. (par défaut : 0, tous les processeurs)
-w	[int]	Taille de mot (Word size). (par défaut : 5)
-orderby	[int]	Trier la liste des résultats par l'un des facteurs suivants, 1 : Conf-score--, 2 : TM-score--, 3 : identité de séquence--, ou 4 : RMSD++, où --/++ signifie ordre décroissant/croissant. (par défaut : 1, Conf-score--) (ignoré dans un alignement par paires)
-mode	[int]	Mode de recherche/alignement, 1: précis, 2: équilibré, 3: rapide, autres valeurs: auto (recherche dans la base de données) ou identique à 1: précis (alignement par paires). (par défaut: auto pour la recherche dans la base de données ; 1 pour l'alignement par paires)
-f	[int]	Activer les filtres mineurs, 0 : désactivé, 1 : activé, ou autre : auto. (par défaut : auto) (toujours 0, désactivé, dans un alignement par paires)

-C	[float]	Seuil de Conf-score (score de confiance). Il doit être entre 0 et 1 ; avec 0, aucun seuil n'est appliqué. (par défaut : 0.5) (toujours désactivé dans un alignement par paires)
-pC	[float]	Seuil de la valeur pC finale, c'est-à-dire -log2(C). Il doit être ≥ 0 ; avec 0, aucun seuil n'est appliqué. (par défaut : 1.0, équivalent à -C = 0.5) (toujours désactivé dans un alignement par paires)
-e	[float]	Seuil de la valeur pC, appliqué à chaque étape de filtrage et de raffinement. Pour les mêmes -e et -pC, -e rejette plus de correspondances non pertinentes. Il doit être ≥ 0 ; avec 0, aucun seuil n'est appliqué. (par défaut : 1.0) (toujours désactivé dans un alignement par paires)
-tmcut	[float]	Seuil de TM-score. Il doit être ≥ 0 ; avec 0, aucun seuil n'est appliqué. Un TM-score ≥ 0.7 par SARST2 pourrait impliquer une homologie au niveau de la famille. (par défaut : 0.15) (toujours désactivé dans un alignement par paires)
-mem	[T/F]	Mettre en cache toutes les données des protéines sujets en mémoire. (par défaut : T) (toujours T, activé, dans un alignement par paires)
-q	[T/F]	Style de sortie rapide. Afficher les résultats dans un format simplifié et facile à analyser. (par défaut : F)
-sa	[T/F]	Afficher l'alignement de séquence basé sur la structure. (par défaut : T)
-mat	[T/F]	Afficher la matrice de transformation pour la superposition. (par défaut : F)
-nmsbj	[T/F]	Normaliser le TM-score par la taille de la structure sujet. (par défaut : F)
-nmavg	[T/F]	Normaliser le TM-score par la taille moyenne de la structure de requête et de chaque structure sujet. (par défaut : F)
-nmusr	[float]	La taille de protéine pour normaliser le TM-score. Elle doit être ≥ à la taille minimale des deux structures ; sinon, le TM-score peut être > 1.

-d	[float]	Le d0 pour la mise à l'échelle du TM-score, par exemple, 5.0 Angströms (Å).
-ml	[T/F]	Appliquer l'apprentissage automatique. (par défaut : T) (toujours F, désactivé, dans un alignement par paires)
-fdp	[str]	Algorithme de programmation dynamique pour les étapes de filtrage. Options prises en charge : NW (Needleman-Wunsch), SW (Smith-Waterman) (par défaut : NW)
-rdp	[str]	Algorithme de programmation dynamique pour l'étape de raffinement. Options prises en charge : NW (Needleman-Wunsch), SW (Smith-Waterman) (par défaut : NW)
-swp	[str]	Chemin vers le fichier d'échange spécifié par l'utilisateur. L'utilisation d'un fichier d'échange peut réduire les coûts de mémoire. (par défaut : aucun)
-Sout	[str]	Dossier de sortie pour les fichiers de superposition de structure. Le dossier sera créé s'il n'existe pas. Le nombre de fichiers de superposition est limité par l'option -detail. (par défaut : aucun)
-html	[str]	Créer un dossier de sortie HTML. Les fichiers de structure superposés seront également générés dans le dossier HTML. (par défaut : aucun)
-jsmol	[str]	Définir le chemin vers le package JavaScript JSmol pour afficher les structures superposées dans la sortie HTML. Il peut s'agir d'un dossier de disque local ou d'une URL HTTP(S). (par défaut : aucun) (URL d'essai : "https://10lab.ceb.nycu.edu.tw/ext/jsmol")
-pssm_out	[str]	Fichier pour stocker les PSSM des codes structurels et de séquence appliqués dans cet algorithme. (par défaut : aucun)
-pssm_pC	[float]	Le seuil de valeur pC pour la construction PSSM. (par défaut : 0.05)
-h		Imprimer le message d'aide (guide rapide).

5.3 Exemple d'utilisation : recherche de similarité structurelle dans une base de données

Rechercher la structure de requête dans une base de données cible préformatée

```
./sarst2 Qry.pdb -db my_db/my_proteins.db -brief 10 -w 7 -e 0.1
./sarst2 Qry.pdb -db my_db/my_proteins.db -brief 10 -d 5.0 -sa F
```

Dans cet exemple, la base de données cible se trouve dans le dossier "my_db", et "my_proteins.db" est le nom de base des fichiers de la base de données cible. Voir le **Manuel : formatdb** pour les instructions sur comment préparer votre propre base de données cible.

5.4 Exemple d'utilisation : alignements un-contre-tous

Rechercher la structure de requête par rapport aux structures sujets listées

```
./sarst2 Qry.pdb Sbj1.pdb Sbj2.cif Sbj3.pdb -mat T
```

Rechercher la structure de requête par rapport aux fichiers de structure sujets spécifiés par des motifs de caractères génériques

```
./sarst2 Qry.pdb "set1/*.pdb" "set2/1a???.cif" -nmavg T
```

Dans cet exemple, "set1/*.pdb" et "set2/1a???.cif" sont entre guillemets et contiennent des caractères génériques. Le programme sarst2 développera automatiquement ces motifs de caractères génériques et récupérera les noms de fichiers correspondants en interne. Si les motifs ne sont pas entre guillemets, le système d'exploitation développera les caractères génériques à la place. Lorsque le nombre de fichiers correspondants est élevé, la liste d'arguments de la ligne de commande résultante peut dépasser les limites du système et entraîner l'échec de la commande. Par conséquent, nous recommandons d'entourer les motifs de caractères génériques de guillemets pour permettre à sarst2 de gérer la liste des fichiers en interne, plutôt que de dépendre du comportement par défaut du système d'exploitation.

Rechercher la structure de requête par rapport à plusieurs dossiers contenant des structures sujets

```
./sarst2 Qry.pdb set1 set2 -nmavg T
```

Dans cet exemple, set1 et set2 sont des dossiers qui peuvent contenir des fichiers de structure protéique. Le programme sarst2 récupérera automatiquement tous les fichiers de ces dossiers (équivalent à set1/* et set2/*). Les fichiers identifiés comme étant au format PDB ou CIF seront sélectionnés et alignés par rapport à la structure de requête.

5.5 Exemple d'utilisation : alignement de structures par paires

Aligner la structure de requête avec une structure sujet

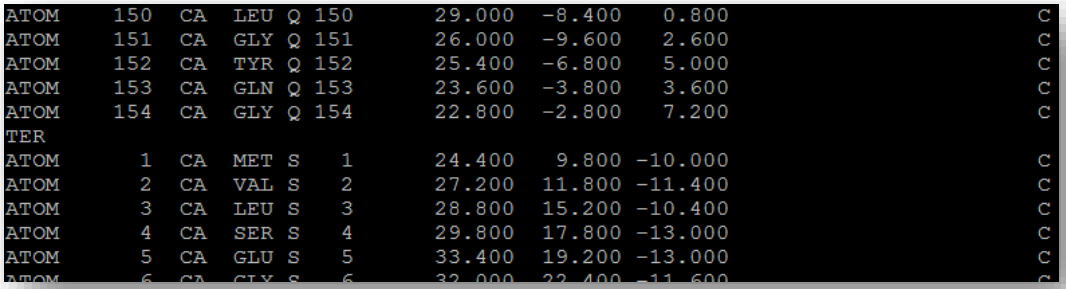
```
./sarst2 Qry.pdb Sbj.pdb -sa F
./sarst2 Qry.pdb Sbj.cif -mat T
```

5.6 Sortie des superpositions structurelles de protéines

Générer des fichiers PDB de superposition structurelle requête-sujet

```
./sarst2 Qry.pdb -db prot/myDb -detail 100 -Sout output_folder
./sarst2 Qry.pdb "set1/*.cif" -detail 100 -Sout output_folder
```

En utilisant l'option -Sout output_folder, les structures protéiques superposées au format PDB seront exportées vers le dossier spécifié par l'utilisateur. Le nombre de structures superposées générées est défini par l'option -detail. Chaque fichier de sortie est nommé Qry-SbjSN.pdb, où SN représente le numéro de série de la protéine sujet dans la liste des résultats. Dans chaque fichier de superposition, les identifiants de chaîne pour les structures des protéines de requête et sujet sont respectivement Q et S. Les deux chaînes sont séparées par un enregistrement TER, comme illustré ci-dessous :



```
ATOM    150  CA  LEU Q 150      29.000  -8.400   0.800                       C
ATOM    151  CA  GLY Q 151      26.000  -9.600   2.600                       C
ATOM    152  CA  TYR Q 152      25.400  -6.800   5.000                       C
ATOM    153  CA  GLN Q 153      23.600  -3.800   3.600                       C
ATOM    154  CA  GLY Q 154      22.800  -2.800   7.200                       C
TER
ATOM      1  CA  MET S   1      24.400   9.800 -10.000                       C
ATOM      2  CA  VAL S   2      27.200  11.800 -11.400                       C
ATOM      3  CA  LEU S   3      28.800  15.200 -10.400                       C
ATOM      4  CA  SER S   4      29.800  17.800 -13.000                       C
ATOM      5  CA  GLU S   5      33.400  19.200 -13.000                       C
ATOM      6  CA  GLY S   6      32.000  22.400 -11.600                       C
```

Comme le montre la figure, seuls les atomes de carbone alpha (Cα) apparaissent dans le fichier de superposition. Ceci est dû au fait que SARST2 effectue tous les calculs basés uniquement sur les coordonnées Cα. L'orientation de la structure de requête reste fixe dans tous les fichiers de superposition, tandis que chaque structure sujet est transformée (pivotée et translatée) en fonction de son alignement avec la structure de requête pour réaliser la superposition.

Pour visualiser les structures superposées, nous recommandons d'utiliser RasMol ou RasWin (http://www.openrasmol.org/). Puisque seuls les atomes Cα sont présents dans les fichiers de superposition, le mode d'affichage dans RasMol doit être défini sur "backbone".

5.7 Générer des pages de résultats HTML interactives

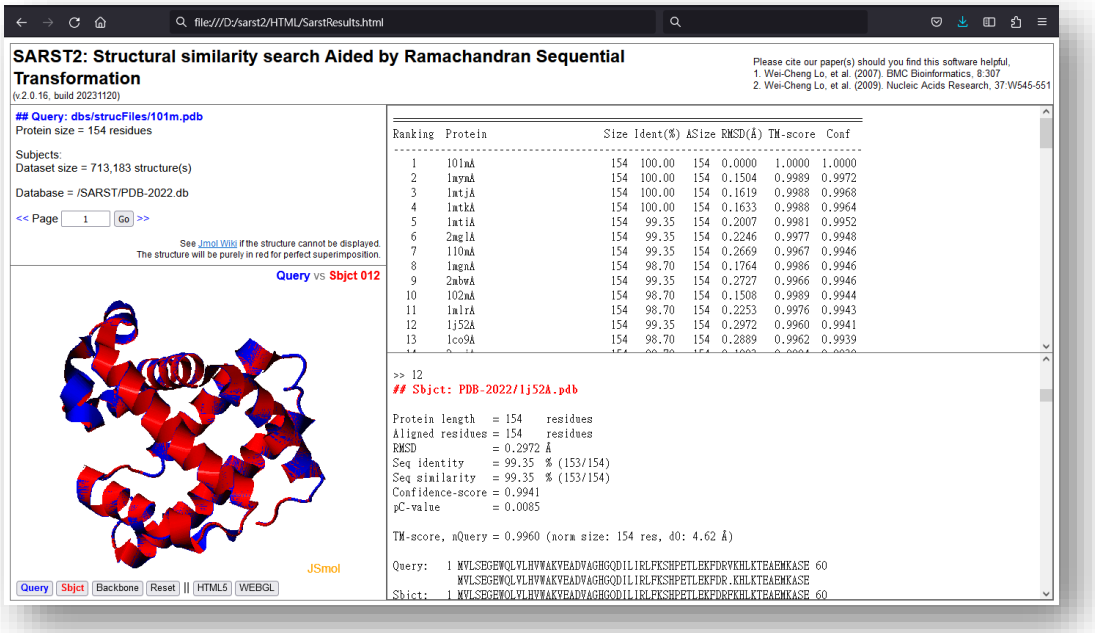
Générer un document HTML avec des scripts JSmol en ligne

```
./sarst2 Qry.pdb -db my_db/my_proteins.db -html output_folder
-jsmol "https://10lab.ceb.nycu.edu.tw/ext/jsmol"
```

Générer un document HTML avec un dossier de script JSmol local (Windows)

```
./sarst2 Qry.pdb "set1/*.cif" -detail 100 -html output_folder
-jsmol file:///D:/software/jsmol
```

En utilisant l'option "-html output_folder", un document de résultats HTML et les fichiers de structure protéique superposés correspondants seront générés dans le dossier spécifié. L'option "-html" doit être utilisée conjointement avec "-jsmol", qui spécifie l'URL ou le chemin local du package JSmol (version 2013). JSmol est un visualiseur interactif de structures moléculaires 3D qui fonctionne dans les navigateurs web et prend en charge la plupart des principaux navigateurs modernes.



Le fichier principal dans le dossier de sortie HTML est SarstResults.html, qui doit être ouvert dans un navigateur web. D'autres fichiers HTML sont intégrés à la page principale à l'aide de cadres internes. Un sous-dossier nommé "sup" sera également créé ; il stocke les fichiers de superposition structurelle entre la protéine de requête et chaque protéine sujet dans la liste des résultats.

Selon votre système d'exploitation, votre navigateur ou votre logiciel antivirus, vous pourriez avoir besoin d'ajuster les paramètres de sécurité pour permettre au navigateur d'exécuter JavaScript et d'accéder aux fichiers de structure superposée dans le dossier "sup", afin que le visualiseur 3D JSmol puisse fonctionner correctement.

6. Manuel : formatdb

6.1 Utilisation

```
./formatdb      structure(s) sujet   -db base de données  [Options]
                ----------------
                > peut être
                1. une liste de fichiers PDB/CIF
                2. des dossiers contenant des fichiers PDB/CIF
                3. un fichier texte brut listant les chemins des
                   fichiers PDB/CIF
```

6.2 Options du programme

-db	[str]	La base de données cible des structures sujettes à créer. (par défaut : aucun)
-flist	[str]	Fichier texte brut listant les chemins des fichiers PDB/CIF. Cet argument peut être utilisé avec les arguments de fichier sujet courants. (par défaut : aucun)
-t	[int]	Nombre de threads. Il doit être ≥ 0 ; avec 0, tous les processeurs seront utilisés. (par défaut : 0, tous les processeurs)
-split	[int]	Diviser la base de données en sous-ensembles, chacun avec le nombre de structures sujettes spécifié par cette option. La division de la base de données aide à empêcher les fichiers de base de données de dépasser la limite de taille de fichier du disque. (par défaut : aucun)
-save_disk	[T/F]	Arrondir les coordonnées des atomes de trois décimales à une décimale pour économiser de l'espace disque. (par défaut : F)
-keep_order	[T/F]	Conserver l'ordre des structures sujettes stockées dans la base de données comme leur ordre d'entrée. La définition de T ralentit la création de la base de données. (par défaut : F)
-h		Afficher le message d'aide (guide rapide).

6.3 Exemples d'utilisation

Créer une base de données cible pour les fichiers de structure sujets listés

```
./formatdb Sbj1.pdb Sbj2.cif Sbj3.pdb -db myDb -keep_order T
```

Plusieurs fichiers de base de données, dont les noms commencent par "myDb", seront créés. L'activation de l'option -keep_order préservera l'ordre des structures sujettes dans la base de données cible selon leur ordre de listage dans les arguments de la ligne de commande.

Créer une base de données cible pour les fichiers de structure sujets listés avec des caractères génériques

```
./formatdb "set1/*.pdb" "set2/*.cif" Sbj1.pdb Sbj2.cif -db myDb
```

Lors de la liste des fichiers sujets, il est possible de mélanger des arguments avec et sans caractères génériques. Il est recommandé d'entourer les arguments avec caractères génériques de guillemets, afin que le programme puisse gérer correctement l'expansion des fichiers.

Créer une base de données cible basée sur une liste de fichiers de structures sujettes

```
./formatdb -flist protlist.txt Sbj1.pdb Sbj2.cif -db myDb
```

Le fichier protlist.txt doit contenir une liste de chemins de fichiers, avec un chemin de fichier par ligne.

Créer une base de données cible à partir de dossiers contenant des fichiers de structure sujettes

```
./formatdb folder1 folder2 -db myDb -save_disk T -split 50000
```

L'activation de l'option -save_disk arrondira les coordonnées Cα à une décimale pour économiser de l'espace de stockage. L'option "-split 50000" entraînera la création de plusieurs bases de données de sous-ensembles, chacune contenant au maximum 50000 structures. Cette option de division est particulièrement utile lorsque la taille des fichiers de base de données formatés peut dépasser la taille maximale de fichier prise en charge par certains systèmes d'exploitation ou formats de disque.

7. Manuel : readdb

7.1 Utilisation

```
./readdb   base de données cible   fichier de sortie   [-seq type_de_séquence]
           ------------------      ---------------     -----------------------
           > base de données       > sera au format        > peut être
             SARST2 préformatée      FASTA                 1. AA
                                                           2. AAT
                                                           3. SARST
                                                           4. SSE
```

7.2 Options du programme

-seq	[str]	Le type de séquence de sortie. AA séquence d'acides aminés AAT séquence de type d'acide aminé à cinq symbols SARST séquence de code de Ramachandran SARST SSE séquence d'éléments de structure secondaire à quatre symboles (par défaut : AA)
-h		Afficher le message d'aide (guide rapide).

7.3 Exemples d'utilisation

Extraire les séquences sujets d'une base de données cible SARST2

```
./readdb my_db/my_proteins.db seqs.fasta
./readdb my_db/my_proteins.db seqs.fasta -seq SARST
./readdb my_db/my_proteins.db seqs.fasta -seq AAT
```

Lorsque l'option -seq n'est pas spécifiée, le type de séquence de sortie par défaut est celui des séquences d'acides aminés. Le fichier de séquences (seqs.fasta) de sortie sera au format FASTA. Si le fichier de sortie existe déjà avant l'exécution de readdb, il sera écrasé.