



Réalisation durant le stage de 1^{ère} année :

Application NPSA-NG

Compétence n°5 : Mettre à disposition des utilisateurs un service informatique

Syméon VACHOT

Contexte

→ Refonte d'une application développée en **Perl** en **1998**

Objectif : Avoir des rendus plus user friendly (*adaptée niveau UX/UI*)



Ancienne interface web du webservice ClustalW (Input)

CLUSTALW

[Abstract] [NPS@ help] [Original server]

Paste a protein sequence databank in Pearson/Fasta format below : help

Output width : 60

CLUSTALW Parameters

Output format : Clustalw

Output order : aligned

Pairwise alignment type : SLOW

Fast pairwise alignment parameters

K-tuple (word) size : 1

Number of top diagonals : 5

Window size : 5

Gap penalty : 3

Scoring method : Percentage

Slow pairwise alignment parameters

Protein weight matrix : GONNET

Gap opening penalty : 10.0

Gap extension penalty : 0.1

Multiple Alignment Parameters

Weight matrix : GONNET

Gap opening penalty : 10.0

Gap extension penalty : 0.2

Residue-specific gap penalties OFF

Hydrophilic gaps OFF

Hydrophilic residues : GPSNDQERK

Percent of identity for delay : 30

Gap separation distance : 8

No end gap separation penalty : ☒

User : public Last modification time : Mon Mar 15 15:24:33 2021. Current time : Sun May 4 10:21:34 2025

Alphafold 2

Options

Model Preset : Monomer

(Only for multimer) # of multimer predictions per model : 5

Model to relax : Best (default)

GPU Relaxation : ☒

DB Preset Configurations : Reduced database (default)

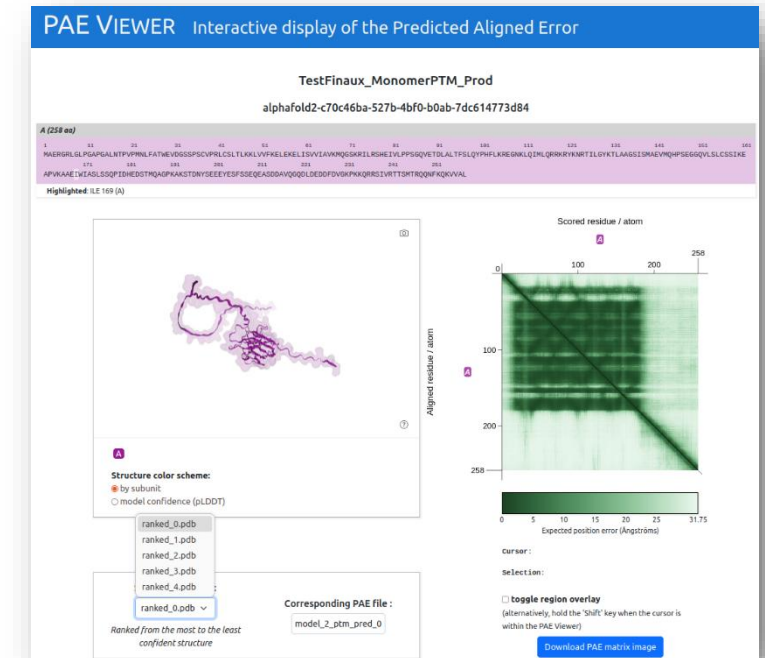
Max template date (YYYY-MM-DD) : 2025-04-01

submit

Every field with a * is required

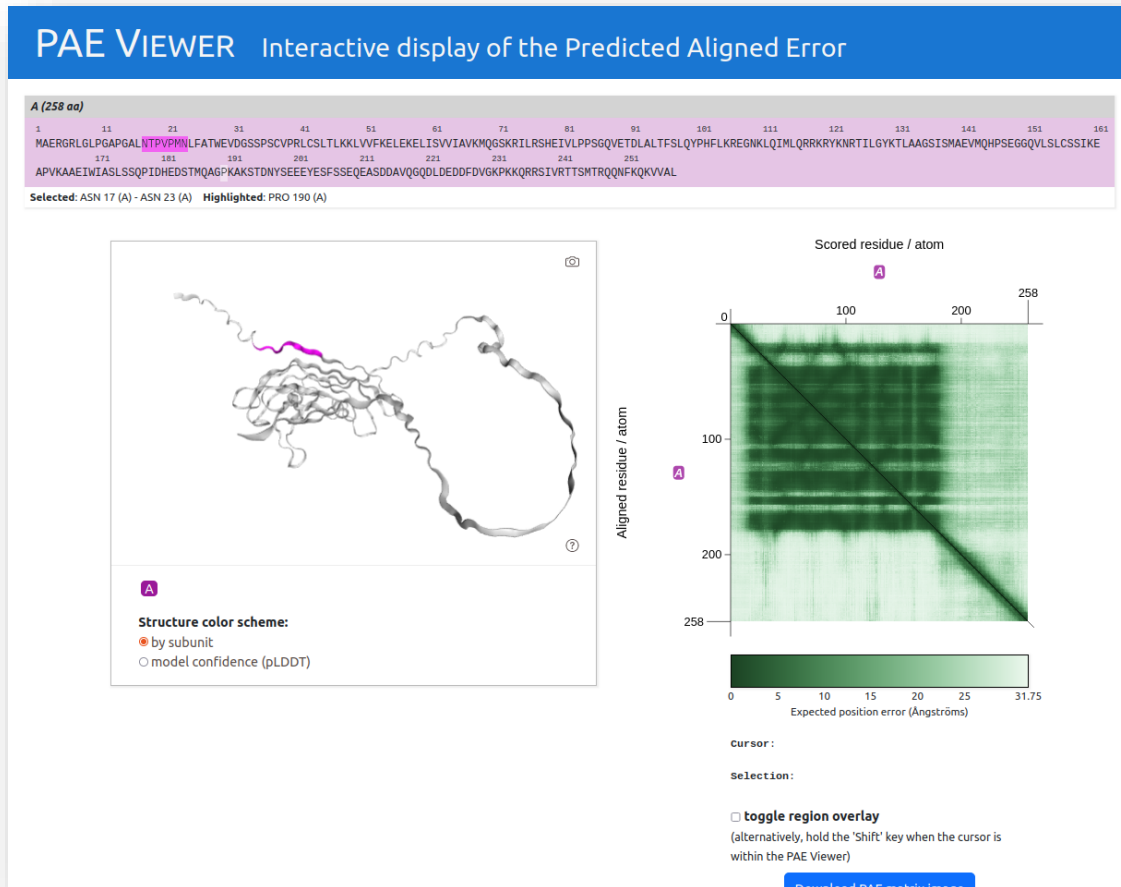
IBCP

Interface web du webservice AlphaFold2 (Input)



Sortie post calcul du webservice AlphaFold2 (Output)

Rôles



→ Déploiement de la solution en production

Objectif : Déploiement final de la solution sur le cluster de calcul

Rendu visuel d'alphafold2 sans options

Rôles

→ Création d'une documentation utilisateur

Objectif : Permettre aux futurs utilisateurs d'appréhender la manipulation du webservice



Documentation_AlphaFold2_NPSA-NG

Documentation Utilisateur : Utilisation du Webservice AlphaFold2 dans NPSA-NG

Introduction

Ce guide présente les étapes pour soumettre une séquence protéique à AlphaFold2 via l'application web **NPSA-NG**, visualiser les résultats, et interagir avec les modèles prédits.

Étape 1 : Remplir les champs d'entrée

Avant de soumettre un job, veuillez renseigner les informations suivantes dans le formulaire :

- **Séquence protéique** : Entrer la séquence au format FASTA.
- **Titre du job** : Nom personnalisé pour retrouver facilement votre tâche.
- **Adresse e-mail** : Pour recevoir les notifications (facultatif selon configuration).
- **Modèles de protéines proposés** : Sélectionner parmi les modèles prédéfinis.
- **Nombre de prédictions souhaitées** : Définir combien de modèles doivent être générés.
- **Autres paramètres** : Options avancées selon vos besoins (ex. : recadrage, base de données, etc.).

Étape 2 : Soumettre le job

Une fois le formulaire complété :

- Cliquez sur **"Submit"** pour lancer l'analyse.
- Une barre de statut vous indique la progression du job :
 - **Queued** : en file d'attente
 - **Running** : en cours de traitement
 - **Finished** : traitement terminé

Étape 3 : Suivre vos jobs

Tous les jobs soumis sont accessibles via la section **"My Jobs"** (dans la barre de navigation principale).

Pour chaque job, vous pouvez voir :

- **UUID** du job
- **Titre** donné au job
- **Webservice utilisé** (ici : AlphaFold2)
- **Date de soumission**
- **Statut actuel** (Queued / Running / Finished)
- **Action possible** : suppression du job

Étape 4 : Visualisation des résultats

En cliquant sur un job dans **"My Jobs"**, vous êtes redirigé vers une page de visualisation détaillée :

- **Titre du job**
- **Visualiseur 3D** de la protéine générée
- **Alignement** entre la structure 3D et la séquence soumise
- **Matrice de confiance PAE** (Predicted Aligned Error) pour estimer la fiabilité de chaque région du modèle

Étape 5 : Interactions avancées

Sur la page de visualisation :

- **Changer de modèle** : via le menu déroulant **"Selected model"**, vous pouvez afficher un autre des modèles générés.
- **Interaction avec la séquence** :
 - Cliquer sur un résidu dans la séquence met en évidence le résidu correspondant dans la **vue 3D**.
- **Interaction avec la matrice de confiance (PAE)** :
 - Cliquer sur une cellule de la matrice met en évidence les acides aminés concernés dans la **séquence** et dans le **viewer 3D**.

Remarques

- Les modèles sont générés via le backend **AlphaFold2**, cela peut prendre plusieurs minutes selon la taille de la protéine et la file d'attente.
- Les données sont stockées temporairement : pensez à les sauvegarder si nécessaire.

Support

Pour tout problème technique ou question fonctionnelle, contactez l'équipe support via l'onglet **"Contact"** ou par mail à : support@npsa-ng.org.

Rôles

→ Création d'une documentation technique

Objectif : *Permettre aux futurs techniciens et développeur de comprendre les choix techniques effectués pour mieux reprendre le projet*

Commande principale de lancement de l'application

Commandes

Backend

installation des paquets

```
source env/bin/activate # entrée dans l'environnement virtuel

pip install [package] # installer un paquet
pip uninstall [package] # désinstaller un paquet
pip install -r requirements.txt # installer les dépendances pour l'application
pip freeze > requirements.txt # mettre à jour le fichier décrivant les dépendances à in
```

lancement de l'api

```
fastapi dev # mode dev, les erreurs sont affichées et l'api n'est accessible qu'en loca
fastapi run # mode production, les erreurs ne sont pas affichées et l'api est accessibl
python app/main.py # lancement de l'api (production) depuis la racine en tant que fichi
```

Frontend

lancement de l'application

```
npm run dev # lancement en mode dev (application disponible qu'en local)
npm run dev -- --host # lancement en mode host (application disponible sur le réseau)
```

production

```
npm run build # transforme l'application en js natif (dossier dist)
node index.js # lancement en production sur un serveur node (dans le dossier dist)
```

FrontendPAE

lancement de l'application

```
python -m http.server 7000
```

Structure de l'environnement de production

Commandes principales SLURM

Commande	Description
squeue	Affiche les jobs en cours pour tous les utilisateurs
sacct -j <jobid>	Affiche le statut et l'historique d'un job
scancel <jobid>	Annule un job en cours
sinfo	Affiche l'état des partitions et des nœuds

Interface utilisateur

Ces machines permettent d'interagir avec les ressources de l'application :

- ws3-dev-npsa-alexis
- ws3-dev-npsa-manu
- ws3-dev-npsa-stagiaire

Connexion possible uniquement en SSH depuis ws3-interface.

Accès aux composants applicatifs

Backend

```
cd /data/dev/npsa-ng-stagiaire/backend/npsa-ng-backend
```

Frontend

```
cd /data/dev/npsa-ng-stagiaire/frontend/npsa-ng-frontend
```

FrontendPAE

```
cd /data/dev/npsa-ng-stagiaire/frontendPAE/npsa-ng-frontend-pae
```

Pour lancer les différentes ressources nécessaires à l'application, voici les commandes shell essentielles : [Commandes](#)

Compétences acquises et difficultés rencontrées



Compétences acquises

- **Déploiement en production**
Premier déploiement du développement à la production
- **Gestion complète d'un projet**
Gestion du projet du développement au design jusqu'aux documentations finales
- **Insertion professionnelle**
Appréhension du déploiement dans un contexte professionnel

Difficultés rencontrées

- Mauvais changement des variables d'environnement
- Blocage de certaines requêtes du reverse proxy

Résultat

Déploiement



Déploiement du
webservice en
production terminé et
fonctionnel lors de mon
départ

Doc. utilisateur



Documentation
utilisateur terminée et
publiée sur GitHub

Doc. technique



Documentation
technique terminée et
publiée sur GitHub