
Projet L2D -Introduction à la bioinformatique

Cahier de recettes

Projet de Programmation

L2D1



Illustration 1: Image de présentation

Projet L2D -Introduction à la bioinformatique

Les informations d'identification du document :

Référence du document :
Version du document : 2
Date du document : 01/05/2021
Auteurs : Mehdi Hamiche Manal Boutajar Adelin Bodnar

Les éléments de vérification du document :

Validé par :
Validé le :
Soumis le :
Type de diffusion :
Confidentialité :

Les éléments d'authentification :

Maître d'ouvrage :	Chef de projet :
Date / Signature :	Date / Signature :

Projet L2D -Introduction à la bioinformatique

Sommaire

<u>1. Introduction</u>	5
1.1. Objectifs et méthodes	5
1.2 Documents de références	9
<u>2. Guide de lecture</u>	10
2.1. Maîtrise d'œuvre	10
<i>2.1.1. Responsable</i>	10
<i>2.1.2. Personnel administratif</i>	10
<i>2.1.3. Personnel technique</i>	10
2.2. Maîtrise d'ouvrage	11
<i>2.2.1. Responsable</i>	11
<i>2.2.2. Personnel administratif</i>	11
<i>2.2.3. Personnel technique</i>	11
<u>3. Description de la fourniture</u>	12
<u>4. Moyens d'essais et outils</u>	13
<u>5. Conformité aux spécifications générales</u>	16
<u>6. Conformité aux spécifications fonctionnelles ou objet</u>	20

Projet L2D -Introduction à la bioinformatique

1. Description	20
2. Procédure de test	23
7. Conformité aux spécifications d'interfaces	24
8. Conformité de la documentation	30
9. Annexes	31
10. Glossaire	34
11. Références	36
12. Index	37

1. Introduction

Ce document aura pour but de décrire le contexte et de compléter le cahier des charges par un ensemble de tests permettant de valider le bon fonctionnement du site du point de vue du client, et en détaillant le plus possible toutes les démarches, c'est-à-dire toutes les idées pensées et discutées avant la conception jusqu'au développement final avec les outils utilisés. Toutes les caractéristiques y seront présentées avec des documents qui illustreront le produit.

Ses tests vont se diviser sur différentes étapes :

- ❖ La remise de l'application et des documents liés au projet ;
- ❖ La vérification de l'ensemble de tests à réaliser sur l'application ;
- ❖ La validation du projet.

1.1. Objectifs et méthodes

- 1) Dans un premier temps, nous avons commencé par penser à l'interface de notre site et à son objectif principal : pouvoir réaliser des alignements de séquences protéiques ou nucléotidiques en utilisant l'algorithme de Needleman & Wunsch, ensuite les afficher sous une interface graphique.

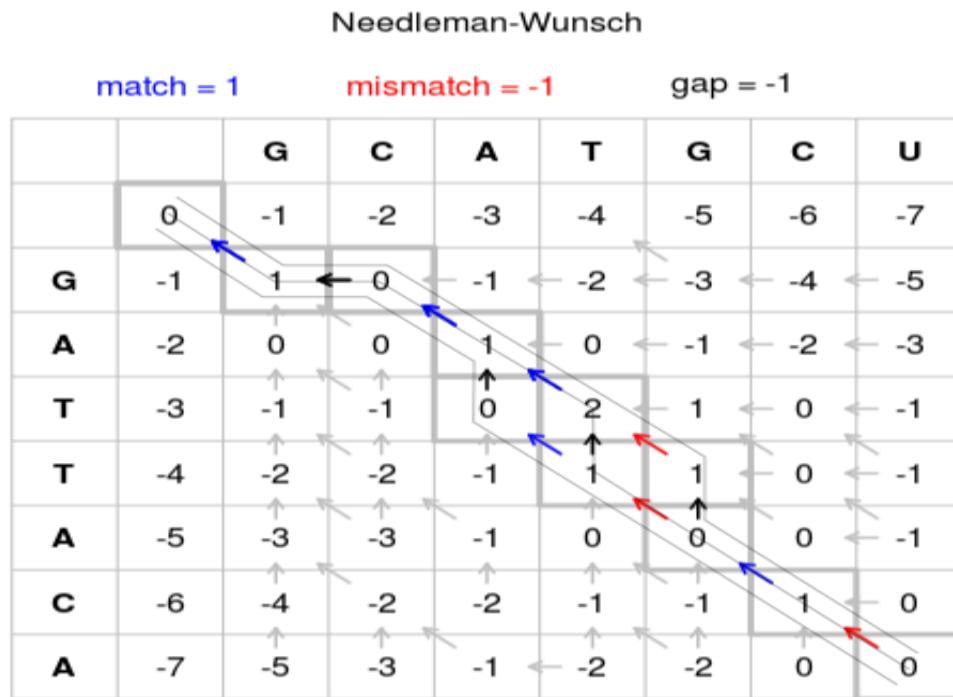


Illustration 2: Schéma explicatif de l'algorithme de Needleman et Wunsch
Alignement Global

- La nature des séquences à comparer : séquences nucléotidiques et séquences protéiques ;
- exemple de l'ADN : ACTG / CTTG (séquences comparée) ;
- exemple de l'ARN : ACUG ;
- exemple de séquences protéiques : >sp|P99999.2|CYC_HUMAN RecName: Full=Cytochrome c

MGDVEKGKKIFIMKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGY
SYTAANKNKGIIWGEDTLMEYLE

NPKKYIPGTMIFVGKIKKEERADLIAYLKKATNE

Projet L2D - Introduction à la bioinformatique

- 2) Nous avons ensuite listé tous les délais donnés qu'on doit respecter (documentation, développement). Pour avoir une vue sur l'interface qu'aura le produit, nous avons réalisé un maquettage à l'aide de figma.
- 3) Une fois la structure du produit mise en tête, nous avons commencé à chercher les technologies les plus adaptées au produit et à nos compétences afin que chacun d'entre nous puisse être le plus à l'aise avec.
- 4) Il ne reste plus que la partie de développement et de mettre à jour au fur et à mesure le cahier des charges et de recettes et l'intégration afin de vérifier si chaque partie sont synergiques, si ce n'est pas le cas, il faudra corriger les erreurs.

Alignment Multiple

Principe

- a) Calcul un score de similarité entre toutes les paires de séquences par comparaison des séquences deux à deux; ensemble de scores d'alignement qui sont regroupés dans une matrice de similarités ;
- b) Cette matrice est utilisée pour trier les séquences, généralement des plus similaires aux plus éloignées ;

Projet L2D - Introduction à la bioinformatique

- c) Cette liste est parcourue itérativement pour construire l'alignement multiple final (pas d'arbre guide) : les deux séquences les plus proches sont alignées (itération 1). A partir de cet alignement, on calcule un «profil» (# une séquence consensus), puis on aligne la troisième séquence avec ce profil (itération 2). Un nouveau profil est calculé avec ces 3 séquences, et la quatrième séquence est alignée.

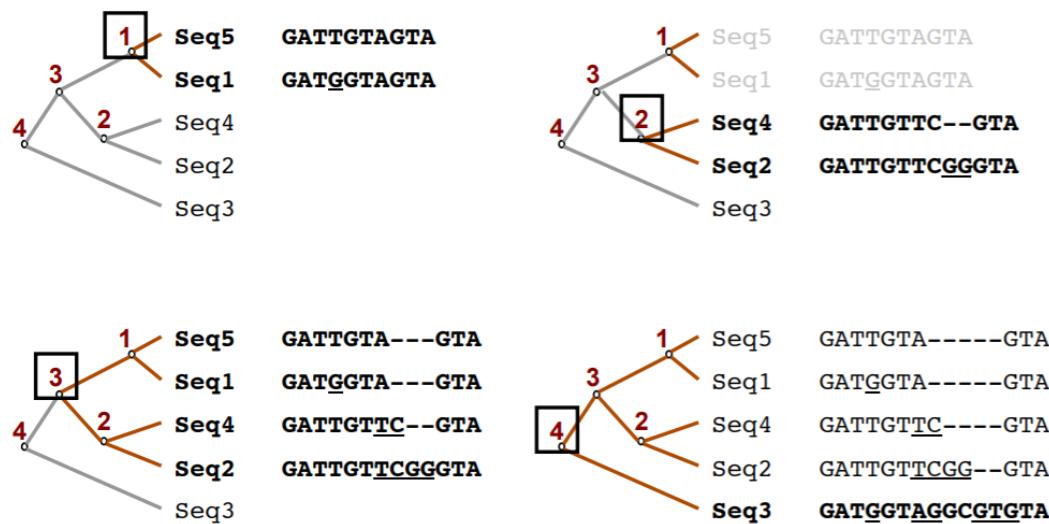


Illustration 3: Exemple d'optimisation d'alignement multiple

- d) DIALIGN est un programme d'alignement multiple qui repose sur une méthode très différente de celle employée par ClustalW. Il s'agit ici d'un algorithme itératif utilisant une approche locale pour calculer les alignements.

Projet L2D - Introduction à la bioinformatique

CLUSTAL format alignment by MAFFT FFT-NS-i (v7.471)

```
sp|P0DTG3.1|AP3 ND--LFNRIFTIGTVTLKQGEIKDATPSDFVRATATIPIQASLPFGNLIVGVALLA---  
sp|P0DTG5.1|VME MA----DSNGTITVEELKK-----LLEQWN----LVZGFLFLTWICL  
sp|P0DTG7.1|NS7 MKIIIILFLALITLTACELYHQE-----CVRGTVLLKEPC  
sp|P0DTG3.1|AP3 --VFOSASKIITLKKRQQLALSKGVHFPVNLLLFVTVVSHLLLVAAGLEAPFLYLYAL  
sp|P0DTG5.1|VME LQFAAYANNRNRFLYIILKLIIFLWLWLPVTACFVL---AAVV-RINWITGGIAAMACLVGL  
sp|P0DTG7.1|NS7 SSGTYEGNSPFMPLADNKF-----ALTCF-----STQFAFACPDPGV  
sp|P0DTG3.1|AP3 VYFLQGINFVRIMRLWLCKRSKPNPLVDANYFLCWHTNCYDVCIPIVNSVTSIVITS  
sp|P0DTG5.1|VME MwLSYfIAfSRL-----FARTRSMSF-----NPENIILLNVPL-H  
sp|P0DTG7.1|NS7 KHV-----YQL-----RARSVSPKLF-----IRQEVE-----  
sp|P0DTG3.1|AP3 GGGTSPISEHDYQIGGYTEK-----WESGVKD---CVVLHSYFTSDYYQLYSTQ  
sp|P0DTG5.1|VME GTILTRPYLESELVIGAVIILRGHLRIGAHHHLGRCDIKDLPEITVATSRTLSYYKLGSQ  
sp|P0DTG7.1|NS7 -QELYSPIF---LVAIAVF-----ITLCFT-----LKRKTE-  
sp|P0DTG3.1|AP3 -LSTDTG-VEHTVFFIYNKIVDEPEEHVOIHTIDGSSGVNPVMEPIYDEPTTTTSVPL  
sp|P0DTG5.1|VME RVAGDSGFAAYSRYRIGN-----YKLNTDHSSSS-----DNIALLVQ---  
sp|P0DTG7.1|NS7 -----LKRKTE-
```

Illustration 4: Alignement multiple au format CLUSTAL

1.2 Documents de références

Les documents du projet servant à l'élaboration du présent document :

- Le cahier des charges qui contient les fonctionnalités de l'application web.
- Le maquettage qui nous permet de modéliser le rendu du produit.

2. Guide de lecture

2.1. Maîtrise d'œuvre

La maîtrise d'œuvre présente l'équipe du développement chargé du bon suivi du cahier de recettes et des besoins dont le maître d'ouvrage fait commande.

Elle représente l'équipe du développement :

- Adelin Bodnar
- Manal Boutajar
- Mehdi Hamiche

Cette équipe veillera au bon suivi du cahier de recettes coordonnées avec le cahier des charges représentant les besoins des enseignants encadrants.

2.1.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'œuvre de lire le document dans sa totalité afin de prendre conscience de l'ensemble des éléments.

2.1.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire la description de la fourniture, conformité aux spécifications générales et la conformité de la documentation.

2.1.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte les parties sur les moyens d'essais et outils, conformité aux spécifications fonctionnelles ou objet et la conformité aux spécifications d'interfaces.

2.2. Maîtrise d'ouvrage

La maîtrise d'ouvrage représente dans notre cas le client du projet, c'est-à-dire les personnes dont les besoins permettent la conception du projet.

La maîtrise d'ouvrage est assistée par l'équipe de la maîtrise d'œuvre et donc ce rôle sera assuré par les enseignants encadrants Dragutin Jastrebic et Koviljka Lukic Jastrebic.

2.2.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'ouvrage de lire le document dans toute sa totalité afin de prendre conscience de l'ensemble des documents.

2.2.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire la description de la fourniture, conformité aux spécifications générales et la conformité de la documentation.

2.2.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte les parties sur les moyens d'essais et outils, conformité aux spécifications fonctionnelles ou objet et la conformité aux spécifications d'interfaces.

3. Description de la fourniture

Le produit final sera livré sous forme d'une application web en état de marche avec son code source. La documentation du produit sera en format numérique directement accessible sur la forge de l'UFR : format **pdf** (.pdf) qui contiendra le manuel d'utilisation et d'installation, le cahier des charges, le cahier de recettes, le plan de tests et d'autres documents nécessaires au projet.

Le code source du produit sera rendu sur le **SVN**.



Illustration 6: Logo de PDF



Illustration 5: Logo SVN

4. Moyens d'essais et outils

Les moyens d'essais et les outils seront à faire avec les tests sur le programme produit à l'aide des environnements de développement destiné au langage java comme Eclipse.

Tableau : structure de données représentant une séquence finie d'éléments auxquels on peut accéder efficacement par leur position, ou indice, dans la séquence. On retrouve les tableaux dans un grand nombre de langages de programmation.

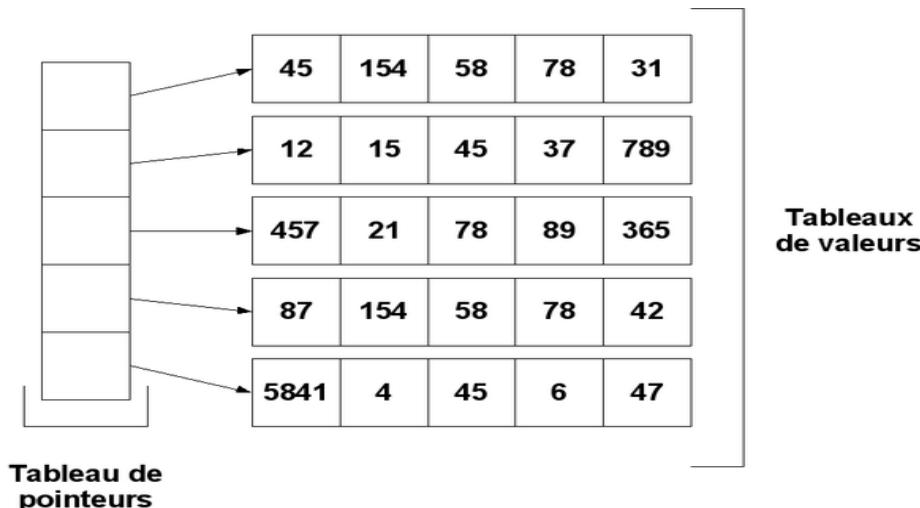


Illustration 7: Tableau à deux dimensions

Langages à typage statique (comme C, Java et OCaml) : éléments d'un tableau - même type

Langages à typage dynamique (APL et Python) permettent des tableaux hétérogènes

Méthode de programmation dynamique :

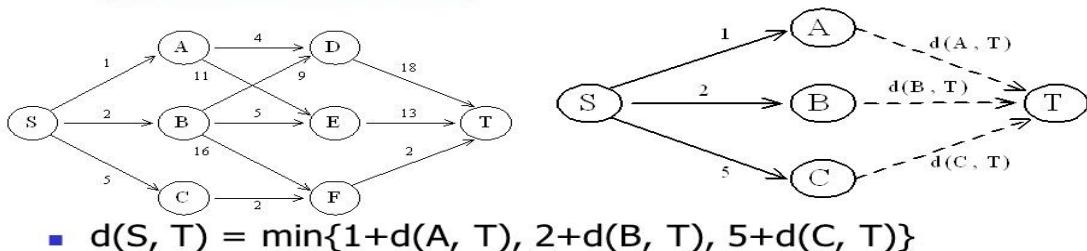
- temps de comparaison de deux séquences de longueur N proportionnel à N^2 ;
- exploration de chaque position de chaque séquence pour la détermination éventuelle d'une insertion augmente d'un facteur $2N$ le temps de calcul ;

Projet L2D - Introduction à la bioinformatique

- méthode qui permet de limiter cette augmentation pour conserver un temps de calcul de l'ordre de N^2 ;
- basé sur le fait que tous les événements sont possibles et calculables mais que la plupart sont rejettés en considérant certains critères. Needleman et Wunsch (1970) ont introduit les premiers ce type d'approche pour un problème biologique.

Approche de programmation dynamique

- (approche vers l'avant) :



$$d(S, T) = \min\{1+d(A, T), 2+d(B, T), 5+d(C, T)\}$$

$$\begin{aligned} d(A, T) &= \min\{4+d(D, T), 11+d(E, T)\} \\ &= \min\{4+18, 11+13\} = 22. \end{aligned}$$

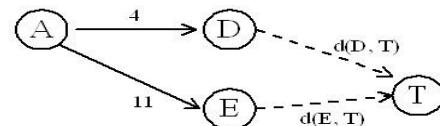


Illustration 8: Schéma explicatif de la programmation dynamique

Algorithme de Needleman et Wunsch :

- Développé pour aligner deux séquences protéiques. Soit A et B deux séquences de longueur m et n. L'algorithme construit un tableau à deux dimensions (m,n) que l'on appelle **matrice** de comparaison.

L'équation suivante résume le principe de calcul d'une case de cette matrice :

$$S(i, j) = se(i, j) + \max((S(i+1, j+1)), (S(x, j+1) - P), (S(i+1, y) - P))$$

Projet L2D - Introduction à la bioinformatique

où $S(i,j)$ est le score somme de la case d'indice i et j , s le score élémentaire de la case d'indice i et j de la matrice initiale et P la pénalité donnée pour une insertion.

- But : trouver le meilleur alignement global, à partir de la matrice transformée.
- Établir dans la matrice un chemin qui correspond au passage des scores sommes les plus élevés, ceci en s'autorisant trois types de mouvements possibles et en prenant comme point de départ le score maximum présent dans la matrice transformée :
 - A. le mouvement diagonal qui correspond au passage de la case (i,j) à la case $(i+1,j+1)$. C'est le mouvement que l'on privilégie.
 - B. le mouvement vertical qui correspond au passage de la case (i,j) à la case $(i,j+1)$, ce qui donne une insertion sur la séquence en i .
 - C. le mouvement horizontal qui correspond au passage de la case (i,j) à la case $(i+1,j)$, ce qui donne une insertion dans la séquence en j .

Dans cet exemple, pas de pénalités pour les insertions, mais possible d'incorporer celles-ci dans la méthode. Pour cela, soustraire dans le calcul de chaque score somme une pénalité en fonction de la position du score "max $S(x,y)$ " considéré.

5. Conformité aux spécifications générales

Présentation générale de tests d'intégration et de tests fonctionnels à faire:

1. Il y a des tests d'intégration de trois modules (Alignement global, Alignement multiple, WEBLOGO) ;
2. Il faut aussi travailler sur l'interactivité de l'interface graphique avec les fichiers en :
 - ➔ testant les boutons (par exemple, liste déroulante pour les identifiants de séquences à aligner),
 - ➔ en appuyant sur un bouton pour chaque alignement et WEBLOGO.
3. Il faut que notre programme affiche les résultats des séquences alignées (tests pour ADN / ARN, tests pour les séquences protéiques) ;
4. Grâce à une fenêtre créée par le module de l'interface graphique , on aurait l'affichage de résultats de l'alignement global ;
5. Grâce à une fenêtre créée par le module l'interface graphique, on aurait l'affichage de résultats de l'alignement multiple;
6. Il faudrait aussi après la réalisation de l'alignement multiple, visualiser le logo – prévue à la connexion au weblogo3.

Tests - détails par module

Alignement global :

- Exemple : "ACTG / CTTG",

Projet L2D - Introduction à la bioinformatique

- Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA",
- cytochromes c,
- gènes (pour cytochromes c),
- séquences nucléiques,
- autres séquences : insuline.

Alignement multiple :

- Exemple : "ACTG / CTTG / CCTG",
- Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA / PPRCLCQRINCI",
- cytochromes c,
- gènes (pour cytochromes c),
- séquences nucléiques,
- autres séquences : insulin, bnlearn_séquences.

Logo :

- Exemple : "ACTG / CTTG / CCTG",
- Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA / PPRCLCQRINCI",
- cytochromes c - trois séquences,
- séquences nucléiques,

Projet L2D - Introduction à la bioinformatique

- autres séquences : insulin, bnlearn_séquences.

Pour ces 4 modules :

● **Interface graphique**

● **Alignement global**

● **Alignement multiple**

● **Logo**

- Avoir les éléments techniques compatibles (Eclipse, Windows ou autres versions) ;
- Tester et chaque fois en cas de réussite - Noter et copier les résultats des tests (image, tableau) ;
- Tester et chaque fois en cas d'échec - Copier les messages d'erreurs - (capture d'écran) ;
- Méthodologie principale : Résultats attendus / Résultats obtenus.

Alignment global

2 séquences v et w:

v : **ATCTGATG** $m = 8$
 w : **TGCATAC** $n = 7$

Alignment : matrice $2 * k$ ($k > m, n$)

v	A	T	--	C	--	T	G	A	T	G
w	--	T	G	C	A	T	--	A	--	C

4 matches 2 insertions 3 deletions 1 mismatch

An introduction de Bioinformatics Algorithms – www.bioalgorithms.info



Illustration 9: Exemple d'alignement global

Illustration 10: Exemple WEBLOGO

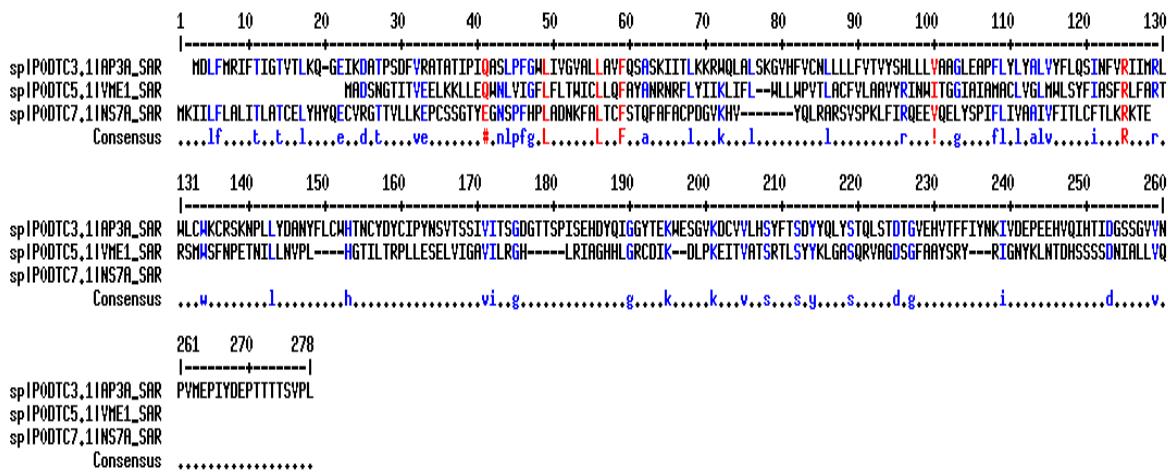


Illustration 11: Exemple d'alignement multiple réalisé MULTALIN

6. Conformité aux spécifications fonctionnelles ou objet

Les différents scenarii de l'application sont décrits sous formes de tableaux contenant les éléments à tester, les contraintes, les dépendances et d'autres informations relatives au scénario.

1. Description

Fonctionnalités attendues :	Résultats qui devront être obtenu :
• Réaliser des alignements globaux	➔ en insérant deux séquences protéiques ou nucléotidiques
• Algorithme de Needleman & Wunsch	➔ appliquer l'algorithme pour aligner les deux séquences
• Construction d'une matrice	➔ avoir la matrice avec les signes de déplacement dans la matrice (flèches) pour le cas : match (identité de deux lettres) et mismatch (deux lettres différentes)
• Initialisation	➔ initialiser les valeurs de la ligne et la colonne avec le gap
• Principe	➔ calculer pour chaque case de la matrice les valeurs selon les cas de mismatch,

Projet L2D - Introduction à la bioinformatique

	match, et gap
<ul style="list-style-type: none">• Répéter chaque fois les démarches	<p>➔ récursion/boucle/condition de fin</p> <p>➔ faire le backtracking (sens opposé par rapport à remplissage) avec l'écriture de chemins possibles de l'alignement global</p>

Projet L2D - Introduction à la bioinformatique

Pour l'alignement multiple :

1. On commence par regrouper les deux séquences les plus proches (groupe 1) ;
2. Regrouper ensuite les groupes les plus proches ;
3. Les deux séquences les plus proches (groupe 2) ;
4. Un groupe avec un groupe (groupe 3) ;
5. Une séquence avec un groupe précédent (groupe 4).

Cet arbre sera ensuite utilisé comme guide pour déterminer l'ordre d'incorporation des séquences dans l'alignement multiple.

On construit un alignement multiple en incorporant progressivement les séquences selon leur ordre de branchement dans l'arbre guide, en remontant des plus proches aux plus éloignées.

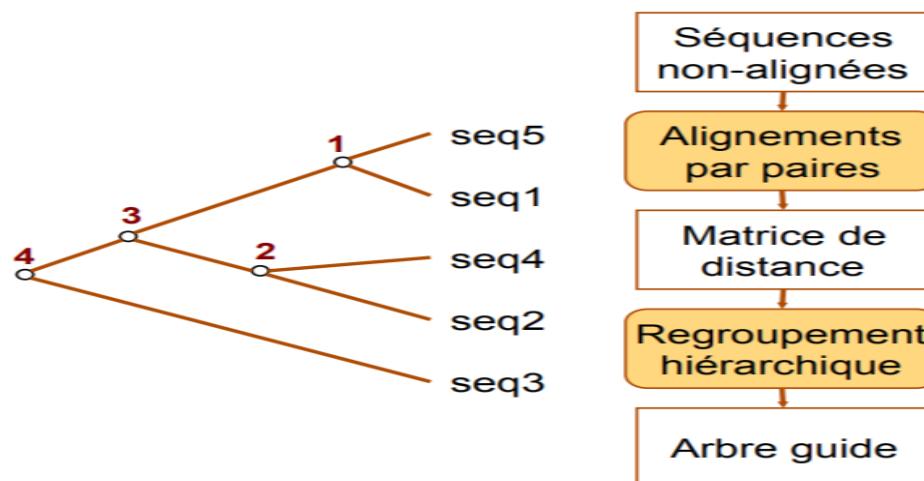


Illustration 12: Schéma explicatif de l'alignement multiple

2. Procédure de test

Avoir la matrice avec les signes de déplacement dans la matrice (flèches) pour le cas :

Projet L2D - Introduction à la bioinformatique

- **match** (identité de deux lettres) et **mismatch** (deux lettres différentes) ;
- Répéter chaque fois les démarches (récursion / boucle / condition de fin) ;
- Faire le **backtracking** (sens opposé par rapport au remplissage) avec l'écriture de chemins possibles de l'alignement global.

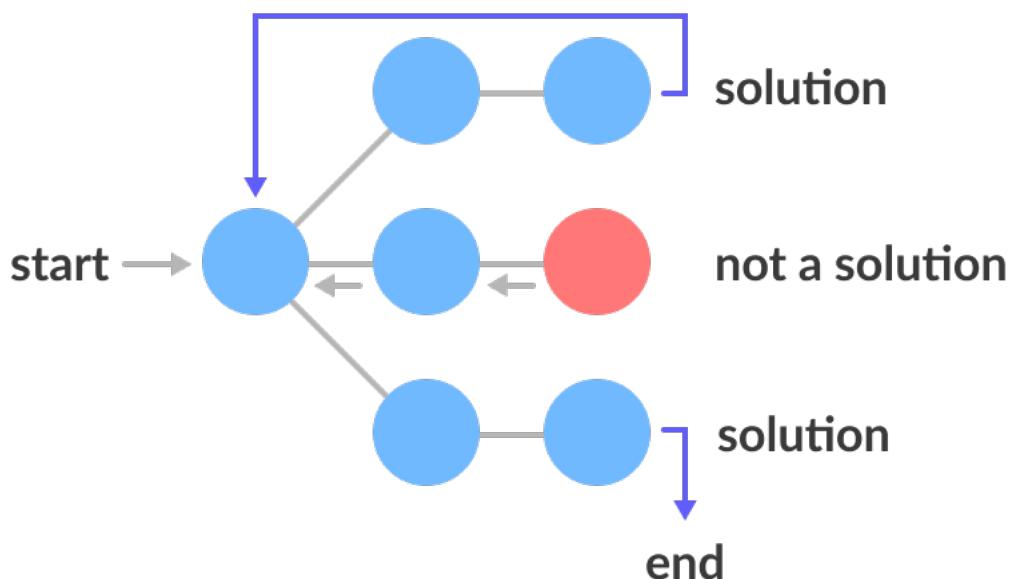


Illustration 13: Schéma explicatif du backtracking

7. Conformité aux spécifications d'interfaces

- Préparer les séquences au format FASTA pour les composants biologiques dans BNLEARN package en R ;
- Continuer à visualiser et à analyser les alignements en python (biotite- Alignement global) et plus tard en R (msa) ;
- Conversion de chaînes de caractères : manipulation de listes / dictionnaires / tableaux avec les séquences à aligner ;
- Réalisation d'alignements utilisant EMBOSS-NEEDLE (Alignement global) et plus tard MAFFT / MULTALIN (Alignement multiple) ;
- Branches : créer selon les langages (JAVA / PYTHON / R / HTML /CSS/) ou selon les modules (Alignement global, Alignement multiple, WEBLOGO, Interface graphique) ;
 - ★ Objectif projet : découvrir la diversité de formats / nature de données biologiques / Bases de données biologiques pour extraire ;
 - ★ les données d'intérêt / outils et programmes bioinformatiques / langages informatiques / Base de données pour stocker les données ;
 - ★ fichiers / protocoles d'échange de données - l'essentiel ;
 - ★ biologiques / voir le circuit d'une donnée biologique : Serveur - Client : server - Base de données biologique - extraction de donnée(s) - client ;

Projet L2D - Introduction à la bioinformatique

- ★ stockage de donnée(s) (fichier / Base de données) - utilisation de donnée(s) > simplification en image.

Interface graphique

1. Première page-boutons, champs, couleur ;
2. Apparition ou ouverture (vitesse : normale, ralentie, inattendue) ;
3. Pages statiques (mise en page : couleur, taille) ;
4. Liens avec le code pour les parties concernées : alignement global, alignement multiple, logo

Alignment global

à vérifier

1. Remplissage de champs à saisie (séquences, match, mismatch, gap) ;
2. Récupération de nouvelles séquences à aligner (stockées dans le(s) fichier(s)) ;
3. Changement de valeurs de gap (choisir les valeurs positives et négatives) ;
4. Affichage des séquences alignées ;
5. Présence du gap (oui, non).

à faire

1. Récupération des séquences à aligner (à partir du fichier) ;
2. Réalisation de l'alignement ;
3. Affichage des séquences alignées ;

Projet L2D - Introduction à la bioinformatique

4. Introduction du gap ;
5. Exemple : "ACTG / CTTG" - alignement (nature nucléique ADN / ARN) ;
6. Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA" - alignement (nature protéique) ;
7. Cas même longueur - 22 séquences déjà envoyées - choisir les séquences à aligner (2 à 2) (faire deux alignements ici) ;
8. cytochromes_c / séquences_nucléiques, / autres - faire l'alignement de séquences 2 à 2 (séquences de 105aa) ;
9. Cas longueur différente - deux alignements.

Alignement multiple

1. Aligner les trois séquences ;
2. Exemple : "ACTG / CTTG / CCTG" ;
3. Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA / PPRCLCQRINCI" ;
4. cytochromes c - trois séquences (minimum) ;
5. cytochromes c / autres séquences pour les tests faire plusieurs alignements multiples changeant le nombre de séquences à aligner, respectant aussi les critères demandés au début (nature, longueur, nombre de séquences)

Logo

1. Exemple : "ACTG / CTTG / CCTG" ;
2. Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA / PPRCLCQRINCI" ;

3. Suivre la dynamique de tests adoptée pour l'alignement multiple ;
 4. cytochromes c - trois séquences (minimum) ;
 5. Afficher les logos de plusieurs alignements changeant le nombre de séquences à aligner, respectant aussi les critères demandés au début (nature, longueur, nombre de séquences).

DONNÉES BIOLOGIQUES :

EXTRAIRE > STOCKER > UTILISER (aligner)



Illustration 14: Logo

R

Illustration 16: Logo Biotite

Illustration 17: Logo

Illustration 18: Logo

8

WONY II

MRLKLTGELRLPKCLNAWQDQLANISMMLFKINIVSNIPFIYEVHELIYIVYIEFPVQCEDMMYYSLYIYGST
|||||::|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:
-MML-EKLNISNTD-PVTPVQIPIEGLTIVYIEFPVQCENM-YEGLCIO-YCT

B

WONV U5
HOJV U5
ORV U5

WONV U5
HOJV U5
ORV U5

MKNLSSVGDGHIWLKIHFLTMGFSINFDPINKFREQTQNINHNINEQLDKLKMWLNGLSH
-----MGFSINFDPIDIKDKFREQQNNHHVNNEQLDKLKMWIWTNFGSQ
-----MGFSINFDPIDIKDGFREQQNNHNGDIDDLQDKLKMWIWTNGLTH
*****:*****:*****:*****:*****:*****:*****:
IKYWFFIIISILTLIFLFLILKTRKLILCKKIFSCCCNVCKKRPKVDIERSKEVKVFSLP
IKYWFLVIIISILMLFIVFLVTKRVLNLCKKIFSCCCFKCRRNKRDRKEDKIVKFSLP
IKYWFLILIIISILVLAILFLILKTRKLILCKKIFSCCCDLCKEKT-SKQRREDKIVKFSLP

Illustration 15: Alignement global EMBOSS-NEEDLE

Projet L2D - Introduction à la bioinformatique

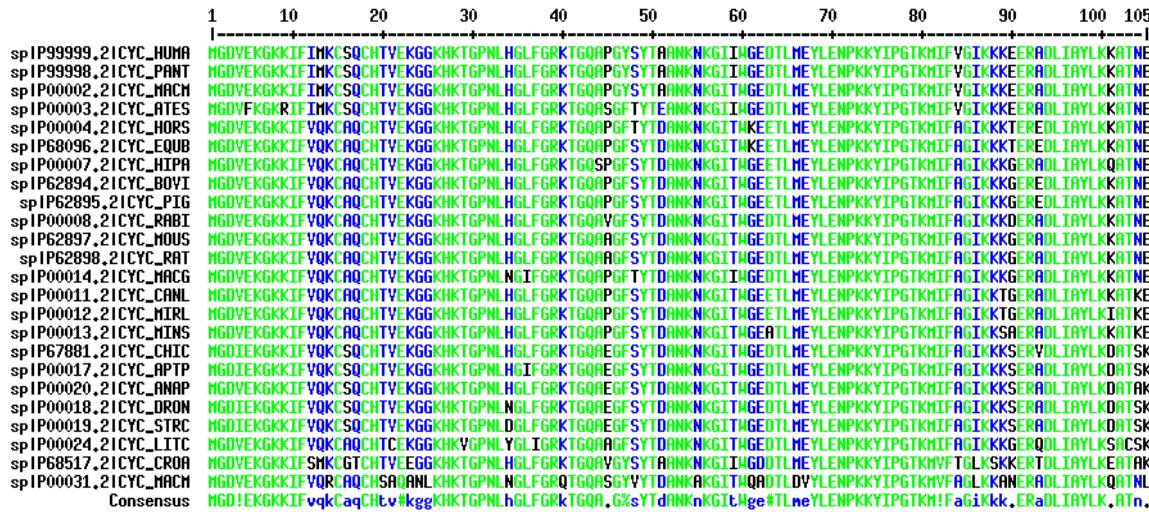


Illustration 19: Alignement multiple avec MULTALIN

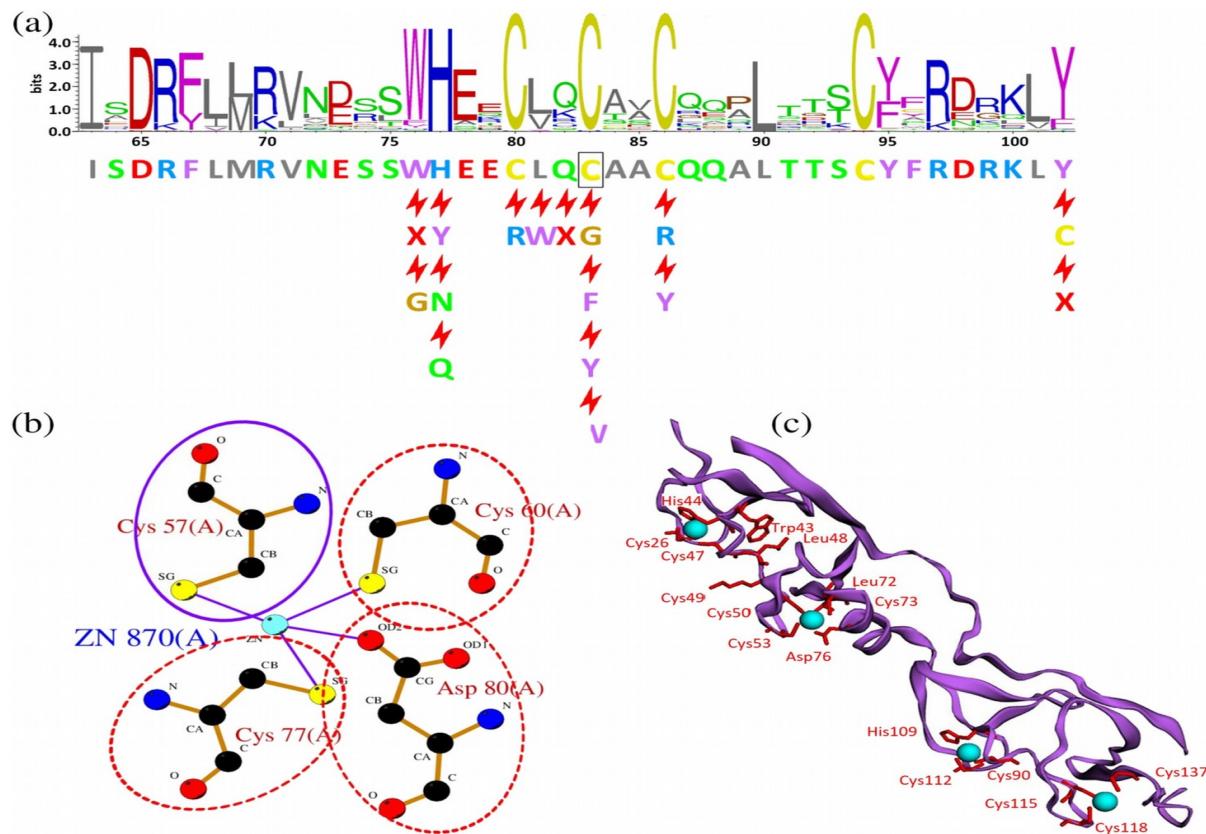


Illustration 20: LOGO PDB

Projet L2D - Introduction à la bioinformatique

- Le produit final est un **programme** capable de faire l'alignement global, l'alignement multiple et le logo,
- Les résultats de l'alignement global, l'alignement multiple et du logo seront présentés via l'interface graphique,
- Le code sera écrit en **JAVA**,
- Les fichiers sont utilisés pour stocker les séquences à aligner,
- Python et R sont utilisés pour visualiser l'alignement global et l'alignement multiple,
- Les bases de données biologiques sont utilisées pour la récupération de séquences.
- Les outils bioinformatiques qui réalisent l'alignement global et l'alignement multiple sont
 - ➔ EMBOSS NEEDLE, MAFFT et MULTALIN,
 - ➔ EMBOSS NEEDLE sert pour l'alignement global,
 - ➔ MAFFT ou MULTALIN sont utilisés pour l'alignement multiple.

8. Conformité de la documentation

Les documents nécessaires liés au projet sont le cahier des charges qui introduit les différentes fonctionnalités de l'application, et les maquettes qui illustrent la vue générale de l'ensemble du produit.

Projet L2D - Introduction à la bioinformatique

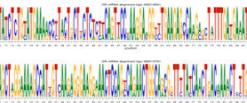
9. Annexes

Cahier des charges et Maquette Figma

Maquette du produit

Groupe L2D1

Description du projet



- ❖ Programme qui analyse et aligne des séquences protéiques,
- ❖ Accéder aux bases biologiques,
- ❖ Manipuler les listes, fonctions, objets, matrices,
- ❖ 3 approches : manuel, programmable et production,
- ❖ Les protéines sont stockées dans des bases de données,
- ❖ Les types d'alignements : global, local, multiple,
- ❖ Affichage des résultats.

Technologie utilisées:

- **HTML** : Ce langage sera utilisé pour définir la structure des pages,
- **CSS** : Ce langage servira à styliser l'ensemble des pages,
- **PHP** : Ce langage permettra la communication entre le client et le serveur.
- **Javascript** : Afin d'animer nos pages web.



- ❖ Travailler sur les parties traitement et visualisation des données - domaine éducatif,
- ❖ Analyse de la structure primaire (amino acids en lettres) des séquences protéiques (cytochrome c et séquences pour les tests) et des séquences nucléiques (génomes) à travers des alignements réalisés,
- ❖ Présenter les résultats des alignements de séquences protéiques et nucléotidiques avec interface graphique.

- **Le cahier des charges** (à rendre semaine 3);
Le cahier de recette (à rendre semaine 4);
La conception générale;
La conception détaillée (à rendre semaine 5);
Le manuel d'utilisation (à rendre semaine 11);
Le manuel d'installation (à rendre semaine 11);
Le plan de tests (à rendre semaine 11);
La documentation interne du code (à rendre semaine 11);
- **Le code sources du programme** (à rendre semaine 11);
Le rapport du projet (avant la soutenance);
Le résumé en français et en anglais (avant la soutenance);
Les diapositives sonorisées (avant la soutenance).

Projet L2D - Introduction à la bioinformatique

Voici aussi les pages statiques du projet en HTML5 et CSS3 :

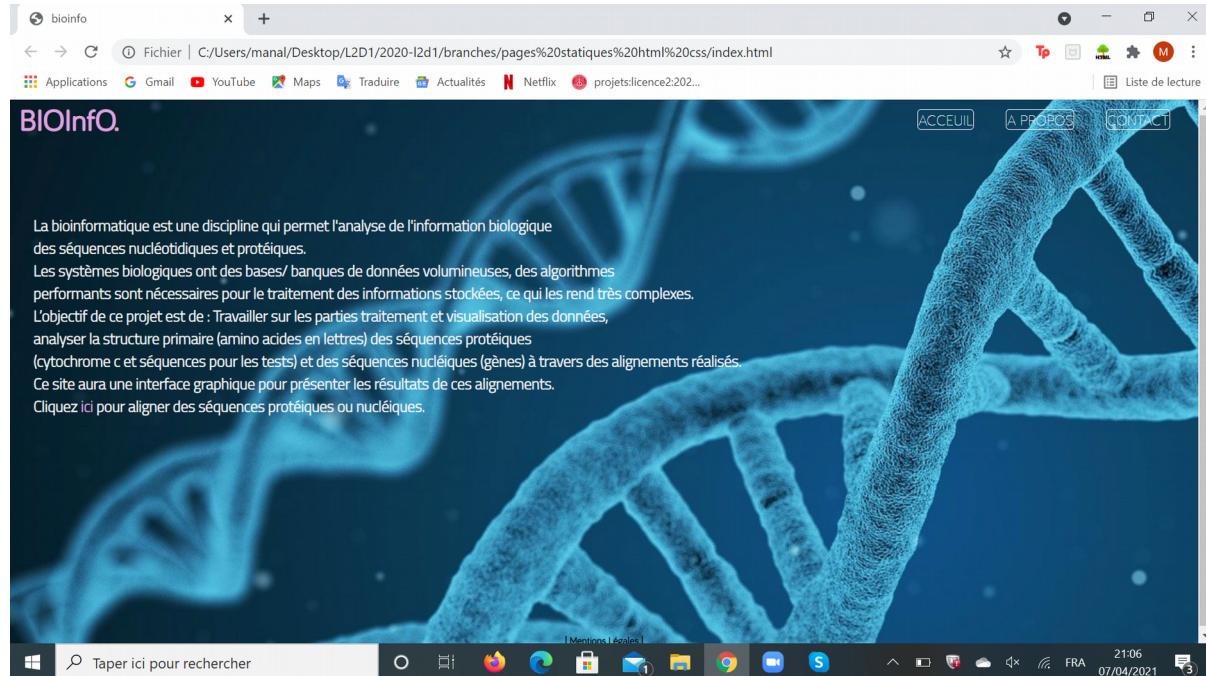


Illustration 21: Page d'accueil HTML5 / CSS3

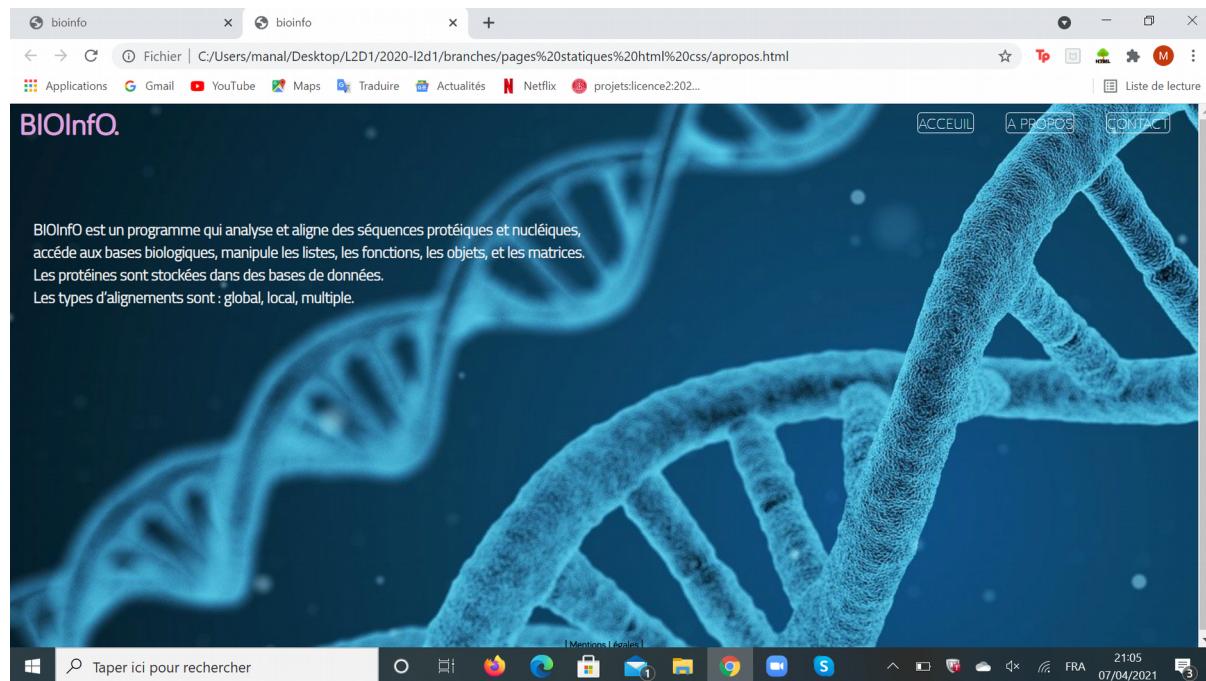


Illustration 22: Page "A propos" HTML5 / CSS3

Projet L2D - Introduction à la bioinformatique

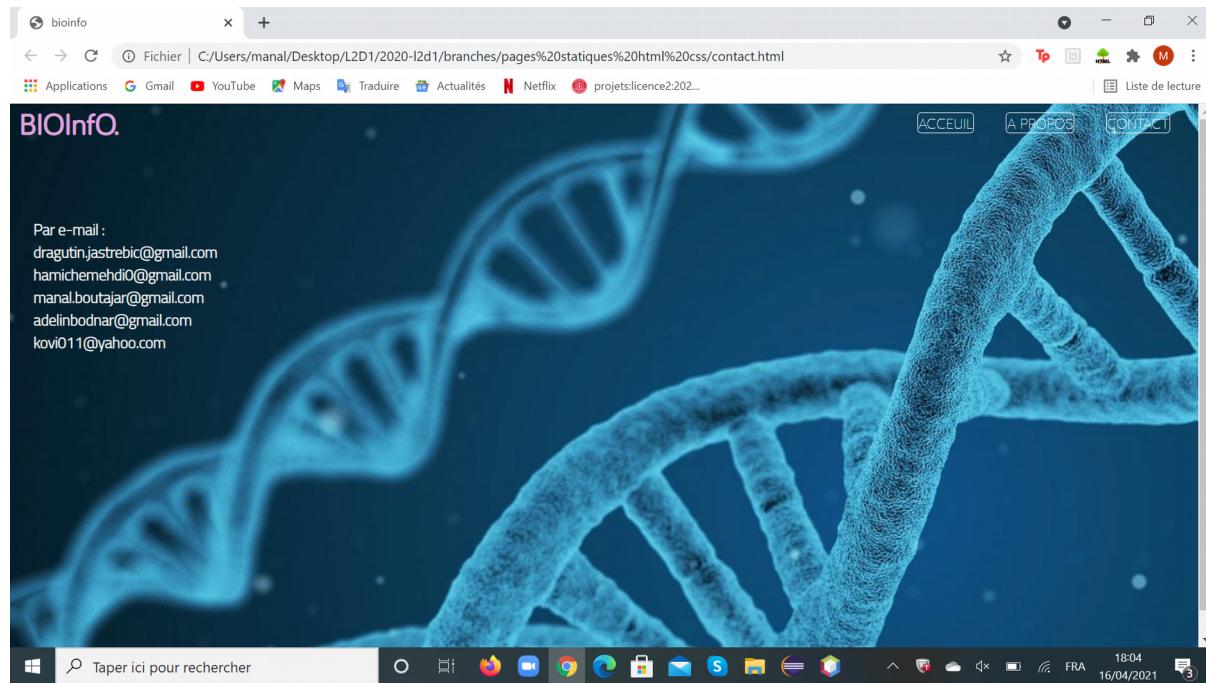


Illustration 23: Page Contact HTML5 / CSS3

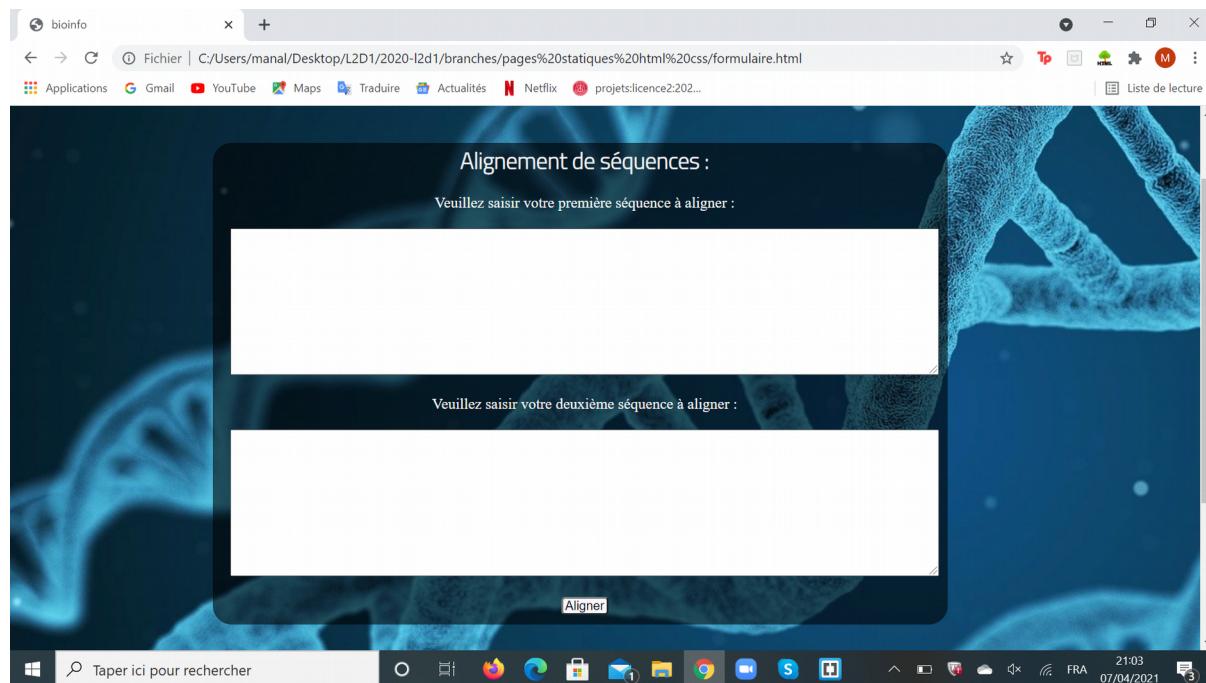


Illustration 24: Page Formulaire HTML5 / CSS3

10. Glossaire

- ★ **BIO-INFORMATIQUE** : discipline permettant d'analyser de l'information biologique qu'il y a dans les séquences nucléotidiques et protéiques
- ★ **PROGRAMMATION ORIENTÉE OBJET** : met en œuvre différents objets (instances de classes). Chaque objet associe des données et des méthodes (fonctions) agissant exclusivement sur les données
- ★ **ALGORITHME NEEDLEMAN - WUNSCH** : effectue un alignement global maximal de deux chaînes de caractères

- ★ **ALIGNEMENT GLOBAL** : recouvre les séquences alignées (par exemple 2 séquences) sur l'ensemble de leur longueur
- ★ **ALIGNEMENT MULTIPLE** : aligner un ensemble de séquences homologues, comme des séquences protéiques ou nucléiques qui assure des fonctions similaires dans différentes espèces vivantes
- ★ **WEBLOGO** : application web conçue pour rendre la génération de logos de séquence aussi simple que possible
- ★ **INTERFACE GRAPHIQUE** : relie les 3 modules et les présente de manière simple et efficace

- ★ **GAP, MATCH, MISMATCH** : valeurs pour alignement global
- ★ **FASTA** : format de fichier texte dans lesquels les séquences sont affichées par une suite de lettres

- ★ **NCBI** : National Centre for Biotechnology Information
- ★ **PROTEIN** : Base de données pour la récupération des séquences protéiques en format .fasta
- ★ **GENE** : Base de données pour récupération des séquences nucléiques en format .fasta

Projet L2D - Introduction à la bioinformatique

- ★ **NUCLEOTIDE** : Base de données pour la récupération des séquences nucléiques en format .fasta
- ★ **UNIPROT** : Base de données pour la récupération des séquences protéiques en format .fasta
- ★ **DDBJ** : Base de données pour la récupération des séquences nucléiques en format .fasta

- ★ **DNA** : Data Bank Japan
- ★ **EMBL** : European Molecular Biology Laboratory

- ★ **EMBOSS** : outil bioinformatique pour l'alignement global - NEEDLE
- ★ **BLAST** : outil bioinformatique pour l'alignement local
- ★ **MULTALIN** : outil bioinformatique pour l'alignement multiple

11. Références

[Alignement de séquences](#)

[BNLEARN](#)

[DDBJ](#)

[National Centre for Biotechnology Information](#)

[Réseaux Bayésiens](#)

[UniProt](#)

[Alignement global](#)

[Alignement multiple](#)

[JAVA - Main \(site 1\)](#)

[JAVA - Main \(site 2\)](#)

[Chaînes de caractères](#)

[Tableau](#)

[Needleman et Wunsch – Java](#)

[Needleman et Wunsch - Python](#)

12. Index

Index des figures

<i>Illustration 1: Image de présentation</i>	1
<i>Illustration 2: Schéma explicatif de l'algorithme de Needleman et Wunsch</i>	6
<i>Illustration 3: Exemple d'optimisation d'alignement multiple</i>	8
<i>Illustration 4: Alignement multiple au format CLUSTAL</i>	9
<i>Illustration 5: Logo SVN</i>	12
<i>Illustration 6: Logo de PDF</i>	12
<i>Illustration 7: Tableau à deux dimensions</i>	13
<i>Illustration 8: Schéma explicatif de la programmation dynamique</i>	14
<i>Illustration 9: Exemple d'alignement global</i>	19
<i>Illustration 10: Exemple WEBLOGO</i>	19
<i>Illustration 11: Exemple d'alignement multiple réalisé MULTALIN</i>	19
<i>Illustration 12: Schéma explicatif de l'alignement multiple</i>	22
<i>Illustration 13: Schéma explicatif du backtracking</i>	23
<i>Illustration 14: Logo R</i>	27
<i>Illustration 15: Alignement global EMBOSS-NEEDLE</i>	27
<i>Illustration 16: Logo Biotite</i>	27

Projet L2D - Introduction à la bioinformatique

<i>Illustration 17: Logo Python</i>	27
<i>Illustration 18: Logo JAVA</i>	27
<i>Illustration 19: Alignement multiple avec MULTALIN</i>	28
<i>Illustration 20: LOGO PDB</i>	28
<i>Illustration 21: Page d'accueil HTML5 / CSS3</i>	32
<i>Illustration 22: Page "A propos" HTML5 / CSS3</i>	32
<i>Illustration 23: Page Contact HTML5 / CSS3</i>	33
<i>Illustration 24: Page Formulaire HTML5 / CSS3</i>	33

Index lexical

algorithme	5, 8, 14, 20
Algorithme	14, 20
alignement	34
ALIGNEMENT	34
alignement global	15, 21, 23, 25, 29, 34 sv
Alignment global	16, 18, 24 sv, 36
Alignment Global	6
ALIGNEMENT GLOBAL	34
alignement global,	29

Projet L2D - Introduction à la bioinformatique

alignement multiple	8, 16, 22, 25, 27, 29, 35
Alignement multiple	16 sv, 24, 26, 36
Alignement Multiple	7
ALIGNEMENT MULTIPLE	34
alignements	5, 8, 20, 24
application web	12
backtracking	21, 23
BNLEARN	24, 36
cahier des charges	5, 7, 9, 12, 30
Cahier des charges	31
CSS	24
DIALIGN	8
EMBOSS NEEDLE	29
EMBOSS-NEEDLE	24
fasta	34 sv
FASTA	24, 34
gap	20 sv, 25
gap	26

Projet L2D - Introduction à la bioinformatique

HTML	24
interface graphique	5, 16
Interface graphique	18, 24 sv
INTERFACE GRAPHIQUE	34
Java	13
JAVA	24, 29, 36
Langages à typage dynamique	13
Langages à typage statique	13
logo	16, 25, 27, 29, 34
Logo	17 sv, 26
LOGO	16, 24, 34
MAFFT	24, 29
maquettage	7, 9
match	20 sv, 23, 25
matrice	7, 14 sv, 20, 23
mismatch	20 sv, 23, 25
mode flash	20
module	16, 24

Projet L2D - Introduction à la bioinformatique

modules	16, 24, 34
MULTALIN	24, 29
MULTALIN :	35
Needleman & Wunsch	5, 20
nucléiques	35
nucléotidiques	5 sv, 20
programmation dynamique	13
python	24
Python	13, 29
PYTHON	24
séquences protéiques	5 sv, 14, 20, 34 sv
séquences protéiques	6
SVN	12
tableau	13 sv, 20, 24
Tableau	13, 36
weblogo	16
WEBLOGO	16, 24, 34
	6, 34 sv