Université Paris Cité

Année universitaire 2022/2023

# Rapport projet court

# Sujet

Assignation des structures secondaire des protéines

Préparé par

Maha GRAA

# Sommaire

# Projet : Assignation des structures secondaire des protéines

## 1) Introduction :

Au sein des cellules, on a trois classes majeurs de polymères qui sont :

- Acide ribonucléique (ARN)
- Acide désoxyribonucléique (ADN)
- Les protéines

C'est à la dernière classe qu'on va s'intéresser durant notre projet. En effet, une protéine est une chaine d'acides aminés.

Les cellules du corps humain font appel à plus de vingt types d'acides aminés différents. On a des acides aminés essentiels pour le corps humain qui sont non synthétisables et qui doivent être apportés grâce a à l'alimentation.

Ces acides aminés sont :

Valine, Leucine, Isoleucine, Lysine, Thréonine, Phénylalanine, méthionine, Histidine, tryptophane, Glutamine, Aspartate, Glutamate, Arginine, Alanine, Proline, Cystéine, Asparagine, Sérine, Glycine, Tyrosine. (Marc parisien. Éd (2005) Les feuillets beta dans les protéines).

### Annotation, comparaison et construction :

Chaque acide aminé est composé d'atomes reliés entre eux par des liaisons covalentes. Ces atomes sont repartis en deux groupes : les atomes de la chaine principale et les atomes de la chaine latérale. Les atomes de la chaine principale sont les mêmes pour tous les acides aminés.

Les acides aminés qu'on retrouve dans les protéines possèdent tous la même structure : qui est un atome de carbone central (α) sur lequel on trouve un groupe carboxyle (COOH) et un groupe aminé (-NH2), un atome d'hydrogène (-H) et un groupement latéral (R). C'est grâce à ce groupement latéral que les acides aminés sont différents l'un de l'autre.

Il existe différents types de liaisons qui s'impliquent dans la structure des protéines : Structure primaire (liaisons peptidiques), secondaire (liaisons hydrogène) tertiaire et quaternaire.

On va s'intéresser en particulier à la structure secondaire, qui est obtenue grâce aux liaisons hydrogènes qui se trouvent au niveau des groupements - CO et – NH des liaisons peptidiques.

On doit se baser sur ces liaisons hydrogène pour pouvoir attribuer la forme de la structure secondaire d'une protéine. Ces structures peuvent être un hélices α (alpha) ou un feuillets β (bêta).

Il existe différentes méthodes pour qu'on puisse définir la structure secondaire citons comme exemple : STRIDE et DEFINE. L'algorithme DSSP (Define Secondary Structure of Proteins) reste la méthode standard pour attribuer une structure secondaire aux acides aminés.

DSSP est un algorithme ayant comme objectif le fait d'approcher la notion intuitive de structure secondaire par un algorithme objectif. Un algorithme qui sert à extraire des caractéristiques structurelles à partir des coordonnées atomiques d'où la reconnaissance des structures (Wiley & Sons.(1983). Dictionary of Protein Secondary Structure:Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Biopolymers, Vol. 22,2577-2637)

## 2) Objectif et problématique :

L'algorithme DSSP est un algorithme standard mais qui n'est pas toujours mis à jour (la dernière mise à jour a eu lieu en 2012).

Notre but durant ce projet est de trouver une méthode alternative à l'algorithme DSSP qui peut attribuer les structures secondaire à une protéine.

## 3) Matériels et méthodes :

Afin de pouvoir créer ce nouveau modèle :

On avait besoin d'installer Python sur notre machine de travail grâce à une installation de miniconda.

En effet Miniconda est une version minimaliste de Anaconda qui embarque Python ainsi que le gestionnaire de packages Conda et grâce à laquelle on peut ensuite installer des packages supplémentaires.

On avait besoin aussi d'un fichier PDB à partir de Protein Data Bank qui est La banque de données sur les protéines du Research Collaboratory for Structural Bioinformatics, plus communément appelée Protein Data Bank. Elle est une collection de données sur la structure tridimensionnelle de macromolécules biologiques qui sont les protéines essentiellement ainsi que les acides nucléiques. Les structures dans Data Bank sont déterminées par cristallographie aux rayons X ou grâce à la spectrométrie RMN. Les données obtenues suite à ces expériences seront déposées dans un fichier PDB. Le Format PDB est le format original de la banque. Ce format est pourtant relativement restrictif, il est en 80 colonnes. Le nombre maximum d'atomes d'un fichier pdb est de 99999. De même le nombre de résidus par chaîne est au maximum de 9999 : il n'y a que 4 colonnes autorisées pour ce chiffre. Le nombre de chaînes, lui, est limité à 62 : une seule colonne est disponible, et les valeurs possibles sont une des lettres des 26 lettres de l'alphabet, en minuscule ou en majuscule, ou un des chiffres de 0 à 9. Quand ce format a été déjà défini, ces limitations n'étaient pas restrictives, mais elles ont plusieurs fois été franchies lors du dépôt de structures extrêmement grandes citons les structures des virus, des ribosomes, des complexes multienzymatiques.

J'ai commencé par choisir une protéine d'intérêt, j'ai cherché dans Protein Data Bank sa structure et j'ai extrait le fichier contenant toutes les informations concernant cette protéine d'intérêt dans un fichier PDB. Pour mon projet, cette protéine est la tyrosine phosphatase yoph from yersina pestis (codée 1huf dans PDB databse). Les liaisons hydrogène ne sont pas citées dans le fichier PDB d'où l'obligation de passer par un programme qui permet de lire CES liaisons.

Parmi les solutions possibles, j'ai installé le programme HBPLUS (HBPLUS v.3.06 - a hydrogen bond calculation program).

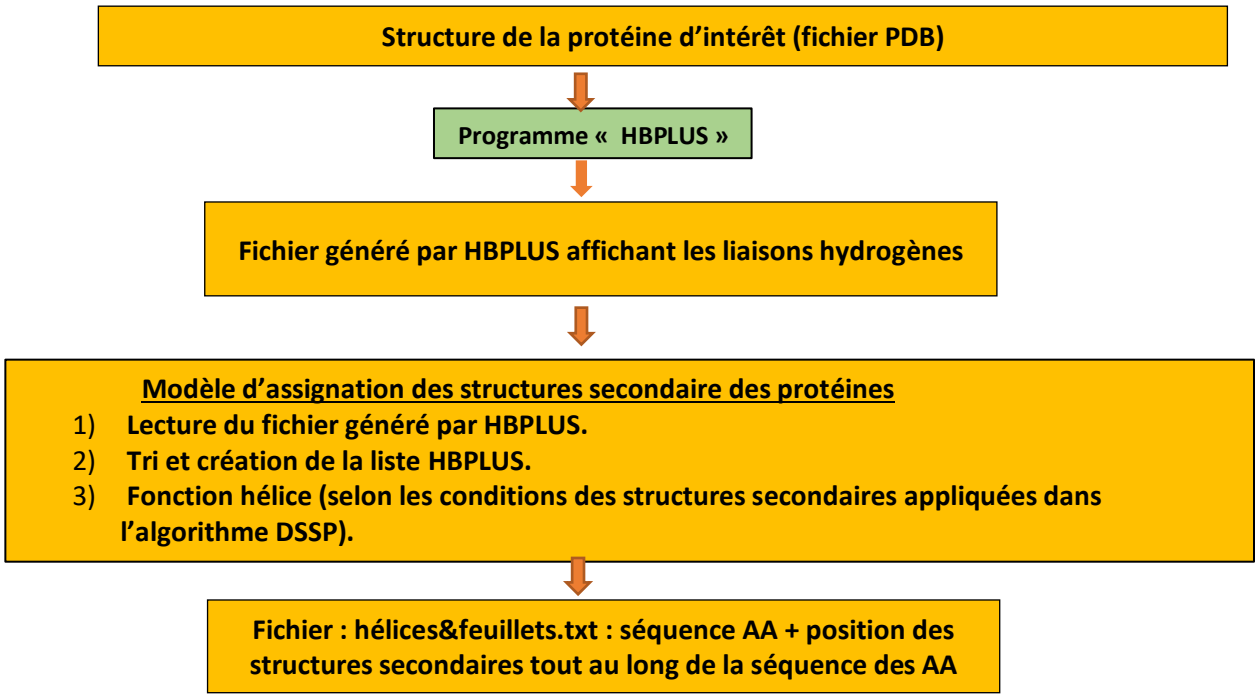
HBPLUS est un programme de calcul de liaisons hydrogène qui permet de calculer les géométries de toutes les liaisons hydrogène éventuellement des listes d'interactions avec les voisins, il calcule également les positions d'hydrogène, traite les hydrogènes qui peuvent occuper plus d'une position, les liaisons H amino-aromatiques, prend en charge la personnalisation complète, par exemple pour les liaisons H il décrit les types d'atomes donneurs et accepteurs.

Il Analyse la liaison H près des chaînes latérales Asn, Gln et His et suggère des conformations optimales

Son fichier de sortie est un fichier PDB comprenant les liaisons hydrogènes et leurs positions.

Ce fichier de sortie me permettra d'avancer dans la réalisation du projet. Une fois le fichier en question est récupéré, on pourra commencer à l'utiliser.

Ci-joint un workflow qui permet de décrire le fonctionnement du modèle :



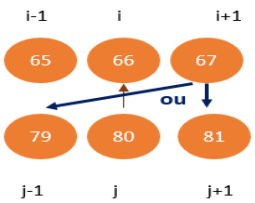

## Description du Workflow (Modèle d'assignation des structures secondaire des protéines) :

Dans cette partie on va décrire le fonctionnement de la pièce centrale du workflow qui est le modèle principal :

1) Lecture du fichier généré par HBPPLUS : pour pouvoir lire ce fichier, on a besoin d'un programme qui permet de lire ligne par ligne le fichier en question (matrice avec une seule colonne et plusieurs lignes) et attribuer les données en question à différentes listes qu'on utilisera ultérieurement.
2) Tri et création de la liste HBPLUS : créer une liste qui est dans ce cas la liste centrale du programme dans laquelle on va stocker toutes nos variables
3) Fonction hélice (selon les conditions des structures secondaire appliquées dans le programme DSSP) : au niveau de cette partie on a choisi d'appliquer les conditions suivies par le modèle standard DSSP, ces conditions sont bien détaillées dans un l'article ( Wiley & Sons.(1983). Dictionary of Protein Secondary Structure:Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Biopolymers, Vol. 22,2577-2637)

Condition hélice : pour avoir une hélice, il faut avoir minimum deux turns de la même catégorie, les turns peuvent être des turns de 3 ,4 ou 5. Cette condition est vrai si seulement si deux acides aminés qui se succèdent sont distants de 3 ,4 ou 5 acides aminés. Une fois cette condition est validée, on vérifie si on a au moins deux turns de la même catégorie qui se succèdent. Si cette condition est validée on peut confirmer que dans patrie de la structure on a une hélice. Il existe trois catégories d'hélices : 3 (hélice $3_{10}$), 4 (hélice α), 5(hélice $\pi$).

4) Fonction feuillet : la condition pour assigner la présence d'un feuillet béta est la suivante : Si on a 3 acides aminés successifs qui ont HBond avec autres 3 acides aminés, la condition qu'on supposera qu'elle est vrai est le fait que l'acide aminé au milieu(i) (voir figure) il a un Hbond avec un autre acide aminé (j) après on passe à la vérification de l'acide aminé suivant (i+1) s'il a un Hbond avec l'acide aminé qui vient juste après (j+1) ou juste avant (j-1) celui qui a un Hbond avec notre condition vraie du début. Ci-dessous une figure qui explique les conditions appliquées.



## 4) Résultats et discussion

En utilisant le modèle mis en place, on aura comme résultat un fichier txt intitulé helices&feuillets qui contient 6 colonnes ([a]position de l'acide aminé, [Type]le nom de l'acide aminé, [hélice]position des hélices, [turn]position et type 3,4, ou 5 des Turn, [bridge]position des bridges,[leader]position des leaders et [feuillet] position des feuillets.

Si on prend comme exemple une protéine modèle « TYROSINE PHOSPHATASE YOPH FROM YERSINIA PESTIS », ayant comme code 1HUF dans la base protein PDB, selon notre modèle, on trouvera 6 hélices et 4 feuillets.

Faisons une matrice de confusion pour comparer le modèle standard DSSP avec notre modèle (sans séparer hélices et feuillets dans la matrice) :

| | |
|---|---|
| True positive (7) | False negative (3) |
| False positive (4) | True negative ? |

On peut constater que sur 10 structures secondaires citées dans le DSSP, on a pu identifier 7 structures secondaire grâce à notre modèle. Notre précision est à 73,33 %

Ainsi on peut constater que parmi 10 structures secondaires retrouvées grâce notre modèle, 7 structures sont vrais.

## 5) Annexes

Exemples d'utilisation des programmes :

### a. Lire fichier hb2 et faire le tri de la liste Hbond_final_tri_num (pièce centrale du script)

```python
#****************** Initialisation ********************************
helices = []
bridges = []
Hbond_pos = []
Hbond_deuxiemepos =  []
type = []
num = []
turn = []
Hbond_final_trie_num = []
num_tri = []
leader = []
feuillet = []
resultat= ''
#****************** lire fichier hb2  ********************************
with open ("1huf.hb2" , "r") as Hydrongen :
    for line in Hydrongen :
        if line.startswith('A0') :
            line = line.split()
            Hbond_pos.append(line[0].split("-")[0][-4:])
            type.append(line[0].split("-")[1])
            Hbond_deuxiemepos.append(line[2].split("-")[0][-4:])
            num.append(line[12])

longueur = len(num)
#****************** tri et création de la liste hbplus **************
i=1
for i in range(longueur) :
    Hbond_final_trie_num+=[ [Hbond_pos[i] , Hbond_deuxiemepos [i]  , type[i]  ]  ]
    helices += ['-']
    bridges += ['-']
    leader  += ['-']
    feuillet += ['-']
Hbond_final_trie_num.sort()

for i in range (longueur) :
    type[i] = Hbond_final_trie_num[i][2]
```

```
(base) PS C:\Users\m_graa\Desktop\projet_Maha_GR> & C:/Users/m_graa/Miniconda3/python.exe c:/Users/m_graa/Desktop/projet_Maha_GR/maha-project.py
[['0003', '2012', 'LEU'], ['0004', '2030', 'SER'], ['0004', '2038', 'SER'], ['0005', '2027', 'LEU'], ['0006', '2009', 'SER'], ['0006', '2089', 'SER'], ['0007', '0004', 'ASP'], ['0008', '0004', 'LEU'], ['0009', '0005', 'HIS'], ['0009', '0118', 'HIS'], ['0009', '
2022', 'HIS'], ['0010', '0006', 'ARG'], ['0010', '0007', 'ARG'], ['0011', '0007', 'GLN'], ['0012', '0008', 'VAL'], ['0013', '0009', 'SER'], ['0013', '0009', 'SER'], ['0013', '0009', 'SER'], ['0014', '0010', 'ARG'], ['0014', '0010', 'ARG'], ['0014', '2021', 'ARG
'], ['0015', '0011', 'LEU'], ['0016', '0012', 'VAL'], ['0017', '0013', 'GLN'], ['0018', '0014', 'GLN'], ['0018', '2078', 'GLN'], ['0019', '0016', 'GLU'], ['0020', '0015', 'SER'], ['0020', '2044', 'SER'], ['0021', '2088', 'GLY'], ['0023', '0020', 'CYS'], ['0023
', '0034', 'CYS'], ['0024', '0052', 'THR'], ['0024', '2093', 'THR'], ['0025', '0108', 'GLY'], ['0026', '0032', 'LYS'], ['0026', '2007', 'LYS'], ['0027', '0051', 'LEU'], ['0030', '2075', 'ASN'], ['0030', '2094', 'ASN'], ['0031', '0011', 'VAL'], ['0032', '0026', '
ALA'], ['0033', '2019', 'ALA'], ['0034', '0024', 'ASN'], ['0034', '0024', 'ASN'], ['0034', '0037', 'ASN'], ['0035', '2049', 'LYS'], ['0036', '0034', 'GLU'], ['0037', '0034', 'THR'], ['0037', '2060', 'THR'], ['0038', '2066', 'THR'], ['0039', '2039', 'PHE'], ['00
40', '0107', 'GLN'], ['0042', '0107', 'LEU'], ['0043', '0046', 'THR'], ['0043', '0046', 'THR'], ['0043', '2056', 'THR'], ['0044', '0106', 'ILE'], ['0045', '2026', 'ALA'], ['0046', '0043', 'SER'], ['0046', '0043', 'SER'], ['0046', '0043', 'SER'], ['0046', '2055
', 'SER'], ['0047', '0044', 'GLY'], ['0048', '0043', 'ALA'], ['0049', '0036', 'ARG'], ['0049', '0052', 'ARG'], ['0049', '2076', 'ARG'], ['0052', '0049', 'GLU'], ['0053', '0049', 'LYS'], ['0053', '2057', 'LYS'], ['0054', '0050', 'VAL'], ['0055', '0051', 'PHE'], [
'0056', '0052', 'ALA'], ['0057', '0053', 'GLN'], ['0057', '2082', 'GLN'], ['0057', '2084', 'GLN'], ['0058', '0054', 'THR'], ['0058', '0054', 'THR'], ['0059', '0055', 'VAL'], ['0060', '0056', 'LEU'], ['0061', '0057', 'SER'], ['0061', '0058', 'SER'], ['0062', '00
58', 'HIS'], ['0062', '0059', 'HIS'], ['0062', '2029', 'HIS'], ['0063', '0060', 'VAL'], ['0065', '2062', 'ASN'], ['0066', '0063', 'VAL'], ['0067', '0123', 'VAL'], ['0068', '0098', 'LEU'], ['0069', '0072', 'THR'], ['0072', '0069', 'ASP'], ['0072', '0069', 'ASP']
, ['0073', '0069', 'THR'], ['0073', '0069', 'THR'], ['0074', '0070', 'ALA'], ['0075', '0071', 'LYS'], ['0075', '0072', 'LYS'], ['0076', '0072', 'LEU'], ['0077', '0073', 'LEU'], ['0078', '0074', 'GLN'], ['0078', '0075', 'GLN'], ['0079', '0075', 'SER'], ['0079',
'0076', 'SER'], ['0079', '0112', 'SER'], ['0080', '0076', 'THR'], ['0081', '0077', 'VAL'], ['0082', '0078', 'LYS'], ['0083', '0079', 'HIS'], ['0083', '2064', 'HIS'], ['0084', '0080', 'ASN'], ['0085', '0080', 'LEU'], ['0087', '0107', 'ASN'], ['0088', '0086', 'TY
R'], ['0088', '0112', 'TYR'], ['0089', '0102', 'ASP'], ['0091', '0099', 'ARG'], ['0093', '0097', 'VAL'], ['0094', '2024', 'GLY'], ['0097', '0094', 'ASN'], ['0098', '0068', 'SER'], ['0098', '0097', 'SER'], ['0099', '0091', 'VAL'], ['0100', '2006', 'LEU'], ['0101
', '0089', 'VAL'], ['0102', '0089', 'SER'], ['0102', '2085', 'SER'], ['0103', '2004', 'LEU'], ['0104', '0087', 'ARG'], ['0104', '0088', 'ARG'], ['0104', '0088', 'ARG'], ['0105', '0087', 'SER'], ['0106', '0042', 'ASP'], ['0107', '0038', 'GLN'], ['0107', '0052',
'GLN'], ['0107', '0105', 'GLN'], ['0108', '2001', 'MET'], ['0109', '2017', 'THR'], ['0109', '2064', 'THR'], ['0110', '0023', 'LEU'], ['0111', '0021', 'GLN'], ['0112', '0109', 'ASP'], ['0113', '0109', 'ALA'], ['0114', '0110', 'LYS'], ['0114', '0110', 'LYS'], ['0
114', '2083', 'LYS'], ['0115', '0111', 'VAL'], ['0116', '0112', 'LEU'], ['0117', '0113', 'LEU'], ['0118', '0114', 'GLU'], ['0118', '0115', 'GLU'], ['0119', '0115', 'ALA'], ['0120', '0116', 'ALA'], ['0121', '0117', 'LEU'], ['0122', '0072', 'ARG'], ['0122', '0072
', 'ARG'], ['0122', '0118', 'ARG'], ['0122', '2074', 'ARG'], ['0123', '0065', 'GLN'], ['0123', '0119', 'GLN'], ['0124', '0120', 'GLU'], ['0125', '0121', 'SER'], ['0125', '0121', 'SER'], ['0125', '0122', 'SER'], ['0125', '0122', 'SER']]
    a      type    helice    turn    bridge    leader  feuillet
```

```
Hbond_final_tri['0125', '0122', 'SER']
```

### b. Fonction hélice

```python
# ******************  Fonction hélice ********************************

for i in range (longueur) :
    a = int (Hbond_final_trie_num[i][0])
    b = int (Hbond_final_trie_num[i][1])
    if abs(a - b)  == 3 :
        turn += [3]
    elif abs(a - b)  == 4 :
        turn += [4]
    elif abs(a - b)  == 5 :
        turn += [5]
    else :
        turn += ['-']

for i in range (longueur):
    if turn[i] == turn[i-1] and turn[i] in [3,4,5]  :
        helices [i] = "H"
        helices [i-1] = "H"
        i=i+1


```

```
helice['-', '-', '-', '-', '-', '-', '-', 'H', 'H', '-', '-', '-', '-', 'H', 'H', 'H', 'H', 'H', 'H', 'H', '-', 'H', 'H
```

## c. Fonction feuillet

```python
#************************Fonction feuillet ******************************
i=1
for i in range (longueur-1) :
    a = int (Hbond_final_trie_num[i][0])
    b = int (Hbond_final_trie_num[i][1])
    a1 = int (Hbond_final_trie_num[i-1][0])
    a2 = int (Hbond_final_trie_num[i+1][0])
    b2 = int (Hbond_final_trie_num[i+1][1])
    if a1<a and a<a2 and a2-a == 1 and a - a1 == 1 :
        if abs (b-b2) == 1 :         # utliser le b de i+1
            bridges [i] = "B"

i=0
for i in range (longueur):
    if bridges[i] == "B" and bridges[i-1]== "B"  :
        leader[i-1] = "L"
        i=i+1
i=0
for i in range (longueur):
    if leader[i] == "L" and leader[i-1]== "L" and feuillet[i-1] == '-' :
        feuillet [i] = "F"
        feuillet[i-1] = "F"
        i=i+2
```

```
feuillet['-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', 'F', 'F', '-', '-', '-', '-', '-', '-',
```

## d. Sortie en terminal

```python
#*********************  Sortie en Terminal  ********************

print("     a      type    helice    turn    bridge    leader  feuillet")
for i in range (longueur) :
    print("    ", Hbond_final_trie_num[i][0], "    ", type[i] ,"    ",helices[i],"       ", turn [i],"       ", bridges[i] ,"    ", leader[i] ,"     ", feuillet[i] )


with open('helices&feuillets.txt', 'w') as f:
    f.write("    a      type    helice   turn    bridge leader feuillet\n")
    for i in range (longueur) :
```

| a | type | helice | turn | bridge | leader | feuillet |
|---|---|---|---|---|---|---|
| 0003 | LEU | - | - | - | - | - |
| 0004 | SER | - | - | - | - | - |
| 0004 | SER | - | - | - | - | - |
| 0005 | LEU | - | - | - | - | - |
| 0006 | SER | - | - | - | - | - |
| 0006 | SER | - | - | - | - | - |
| 0007 | ASP | - | 3 | - | - | - |
| 0008 | LEU | H | 4 | B | - | - |
| 0009 | HIS | H | 4 | - | - | - |
| 0009 | HIS | - | - | - | - | - |
| 0009 | HIS | - | - | - | - | - |
| 0010 | ARG | - | 4 | - | - | - |
| 0010 | ARG | - | 3 | - | - | - |
| 0011 | GLN | H | 4 | B | L | - |
| 0012 | VAL | H | 4 | B | - | - |
| 0013 | SER | H | 4 | - | - | - |
| 0013 | SER | H | 4 | - | - | - |
| 0013 | SER | H | 4 | - | - | - |
| 0014 | ARG | H | 4 | - | - | - |
| 0014 | ARG | H | 4 | - | - | - |
| 0014 | ARG | - | - | - | - | - |
| 0015 | LEU | H | 4 | B | L | F |
| 0016 | VAL | H | 4 | B | L | F |
| 0017 | GLN | H | 4 | B | - | - |
| 0018 | GLN | H | 4 | - | - | - |
| 0018 | GLN | - | - | - | - | - |
| 0019 | GLU | - | 3 | B | - | - |
| 0020 | SER | - | 5 | - | - | - |
| 0020 | SER | - | - | - | - | - |