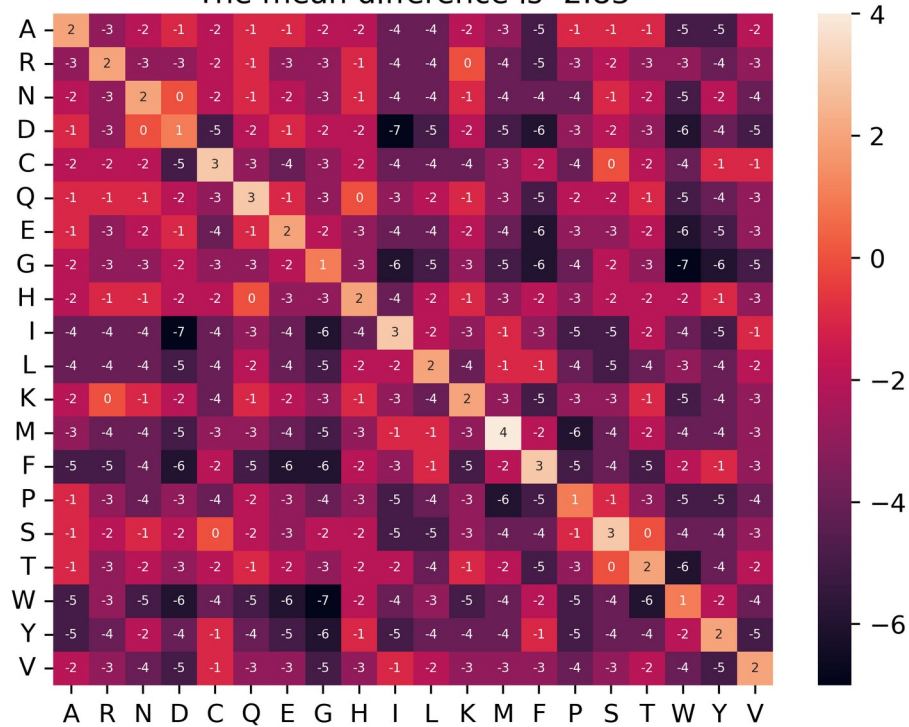


Recalcul de Blosom avec pid et clustering
corrigés

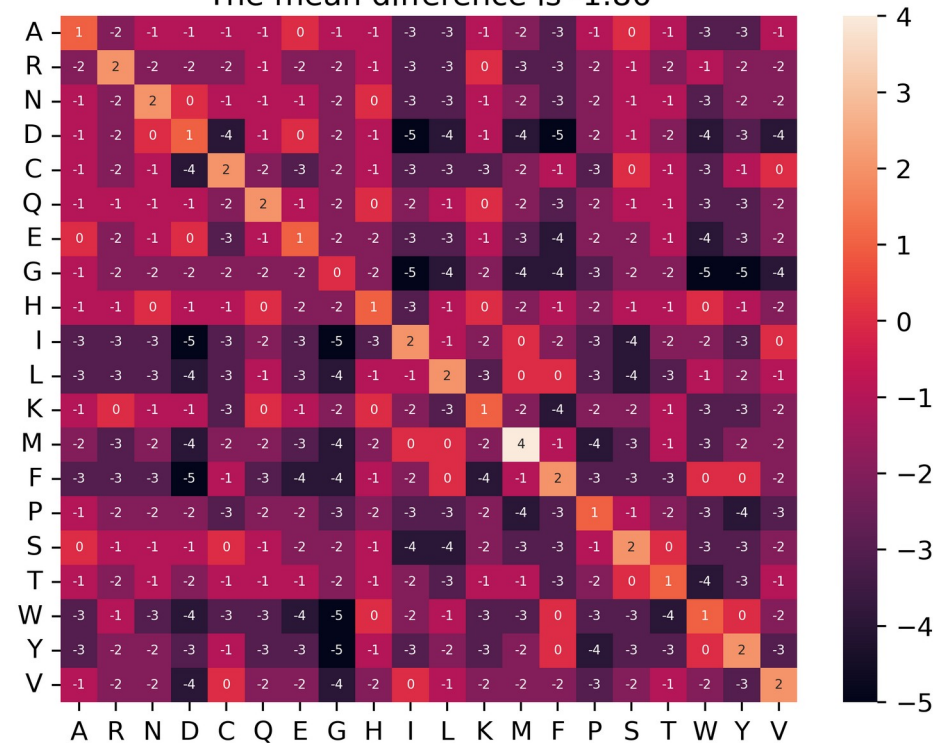
Variante blosum62 (5min)

Heatmap of the difference in Score between Blosum(Pfam_train)
and Blosum62Ref
The mean difference is -2.83



Variante blosum50 (20min)

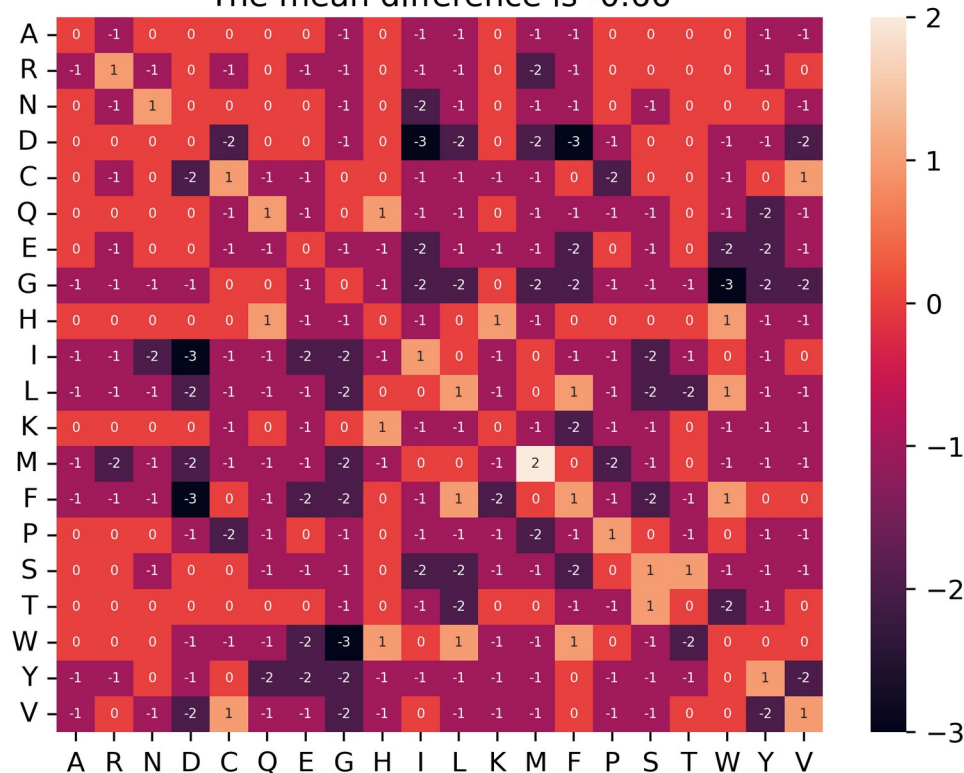
Heatmap of the difference in Score between Blosum(Pfam_train)
and Blosum62Ref
The mean difference is -1.86



Variante blosum30 (1,5h)

Heatmap of the difference in Score between Blosum(Pfam_train)
and Blosum62Ref

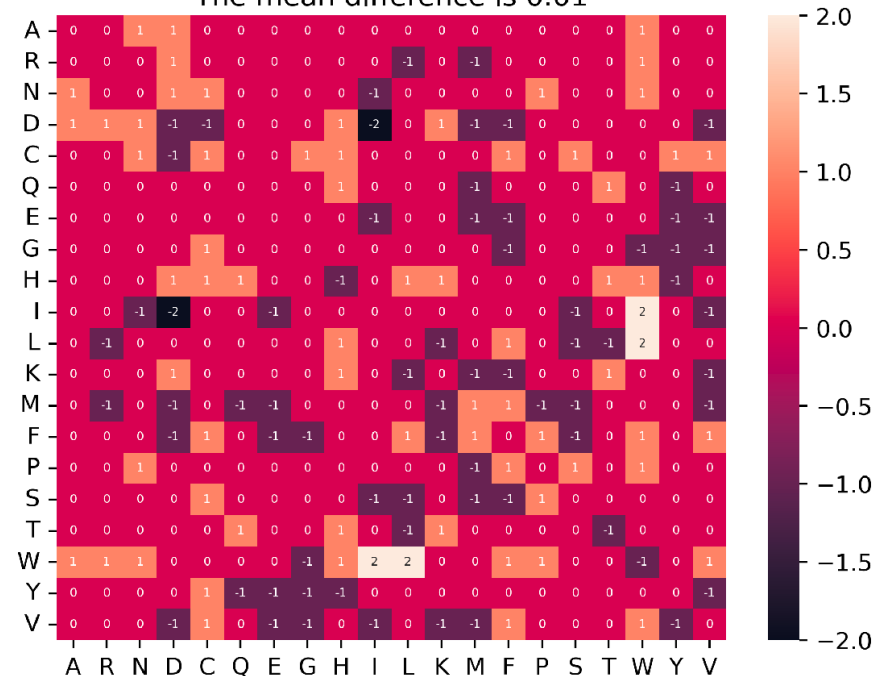
The mean difference is -0.66



Variante blosum0 (3,7h)

Heatmap of the difference in Score between Blosum(Pfam_train)
and Blosum62Ref

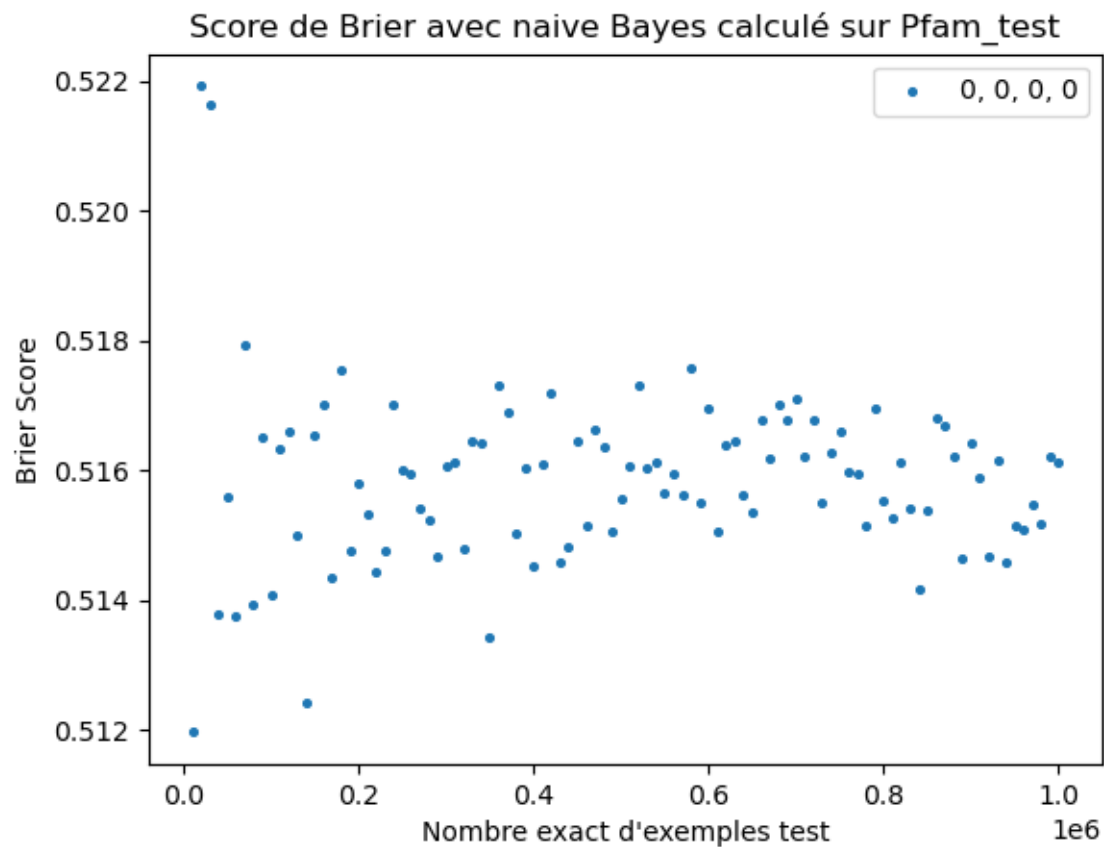
The mean difference is 0.01



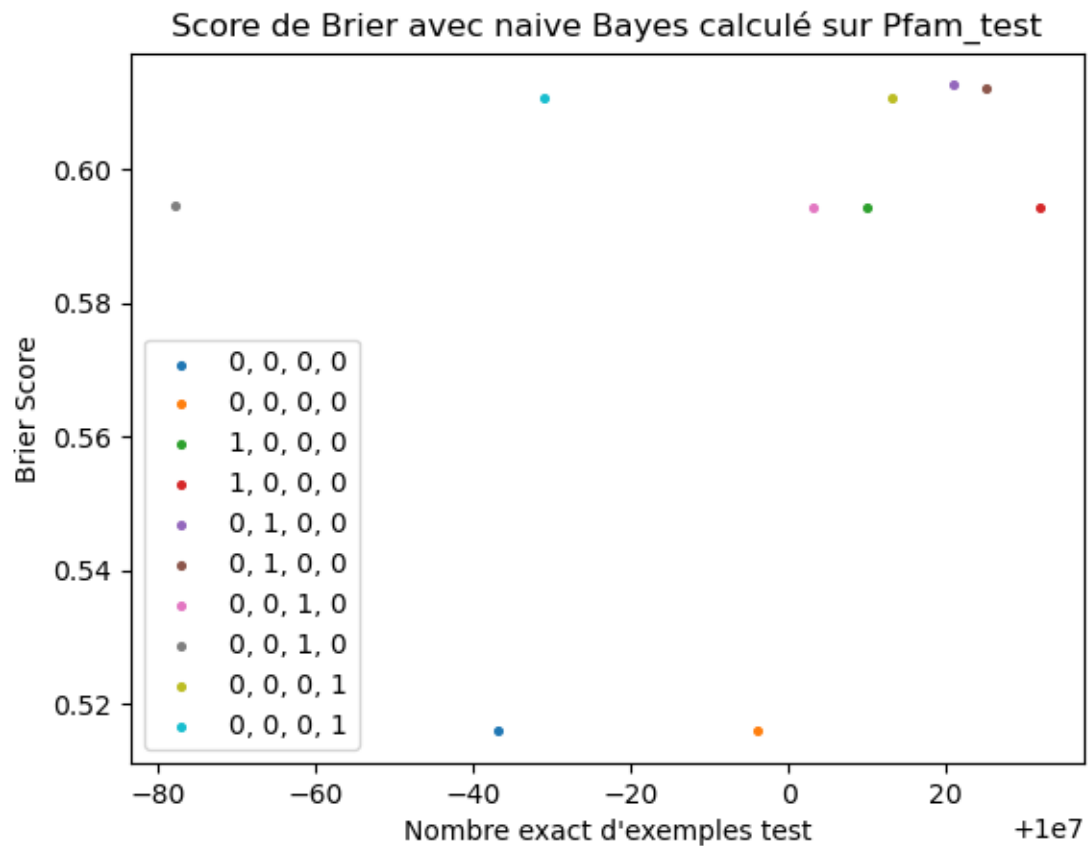
Recalcul de Scores de Brier avec Bayes Naif

(avec pid = 62, temps de calcul 13min/cube)

Stabilisation du Score (test de 10m à 1M d'exemples)



Sur 10M d'exemples demandés



Tests préliminaires supplémentaires

seeds identité

```
>seq1  
ARNDCQEGHILKMFPSTWYV  
>seq2  
ARNDCQEGHILKMFPSTWYV
```

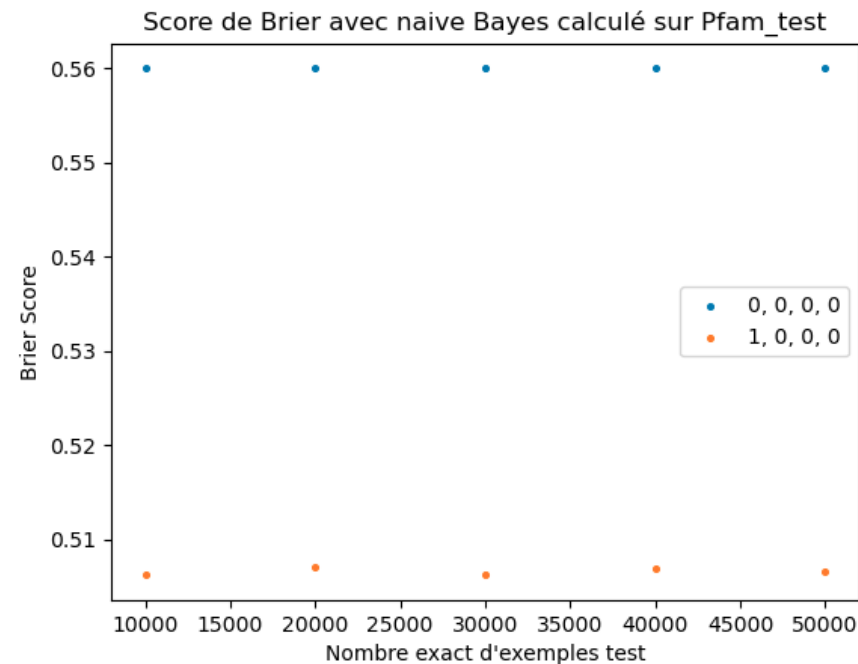
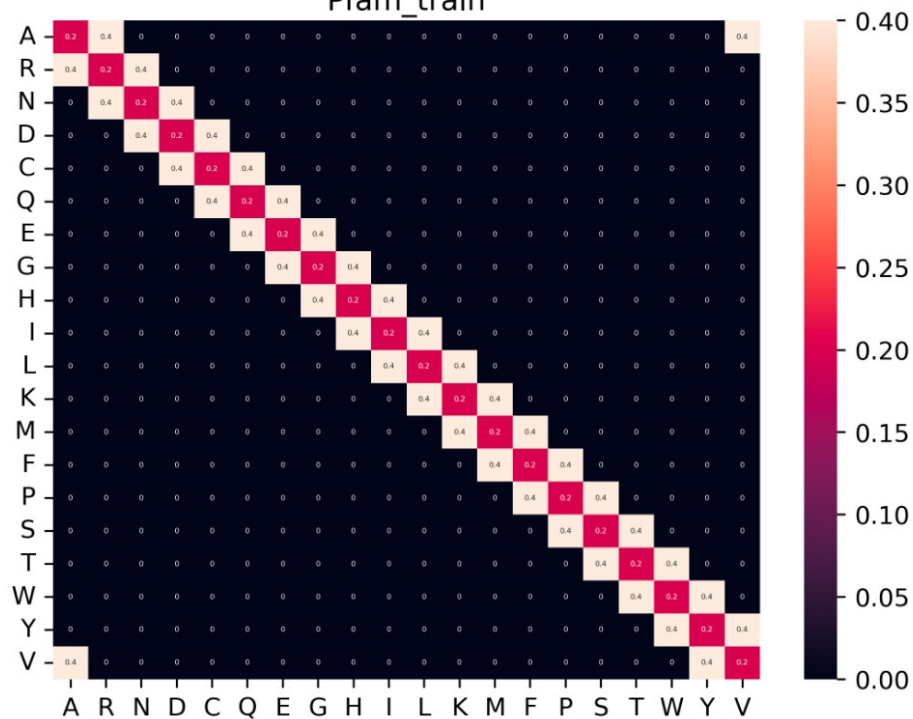
seeds décalés

```
>seq1  
ARNDCQEGHILKMFPSTWYV  
>seq2  
RND CQEGHILKMFPSTWYVA
```

- 10 seeds identité/décalés (5 train, 5 test)
- Pas de clustering 99 %
- pid_inf = 0
- Imposer ordre seq1, seq2

1 identité, 9 décalés (id dans le train)

Heatmap of the conditional probability matrix computed on Pfam_train

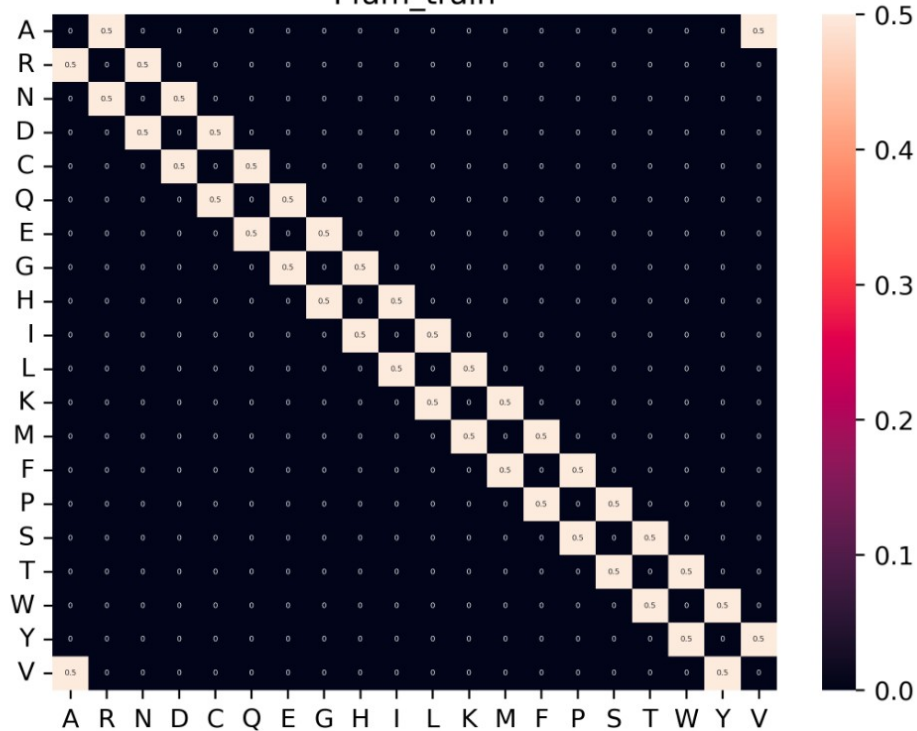


0-0- ... -0-0,4-0,2-0,4-0-0 ...0

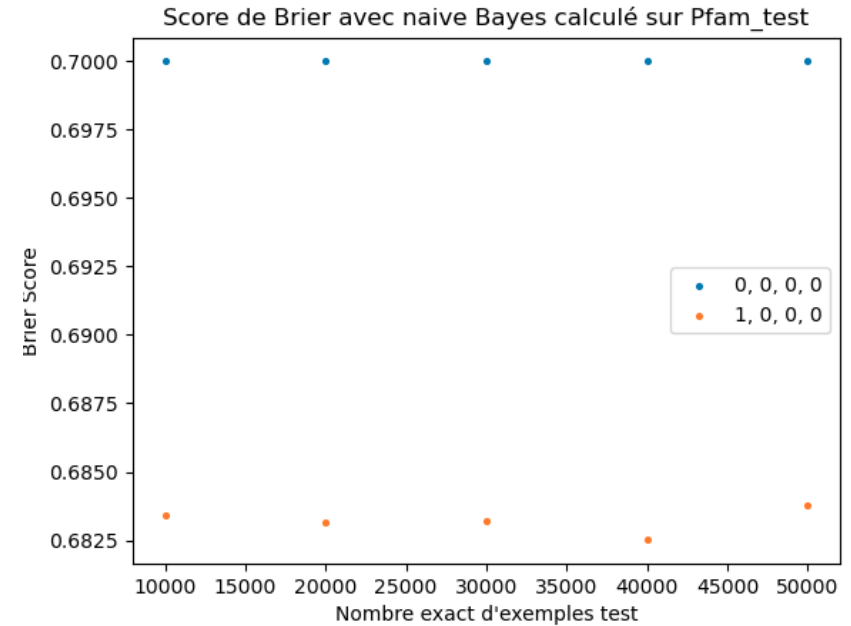
1 identité, 9 décalés
(id pas dans le train)

CAS 2

Heatmap of the conditional probability matrix computed on
Pfam_train



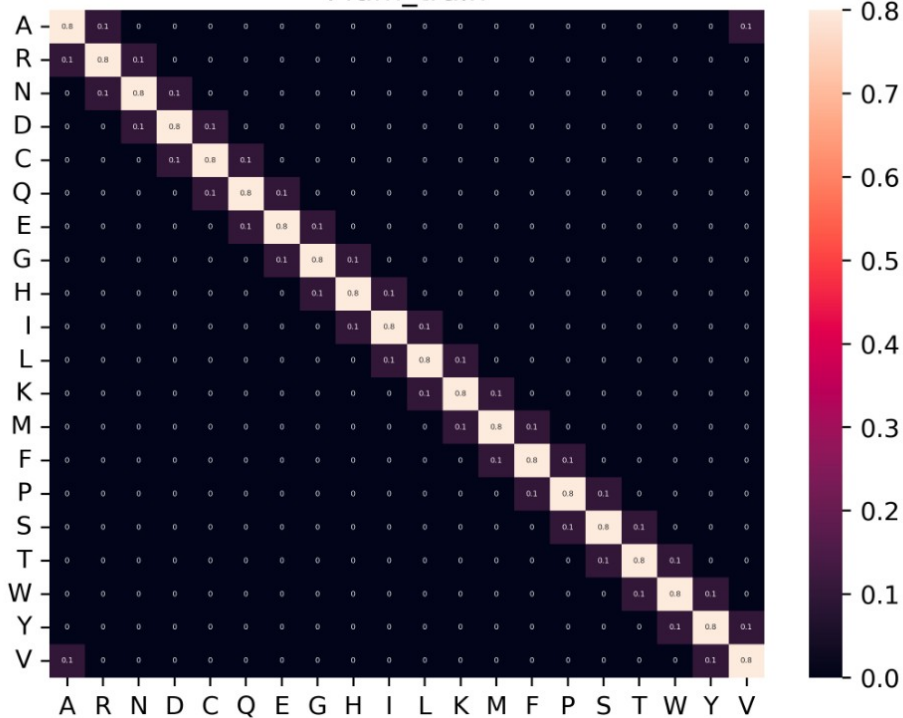
0-0- ... -0-0,5-0-0,5-0-0 ...0



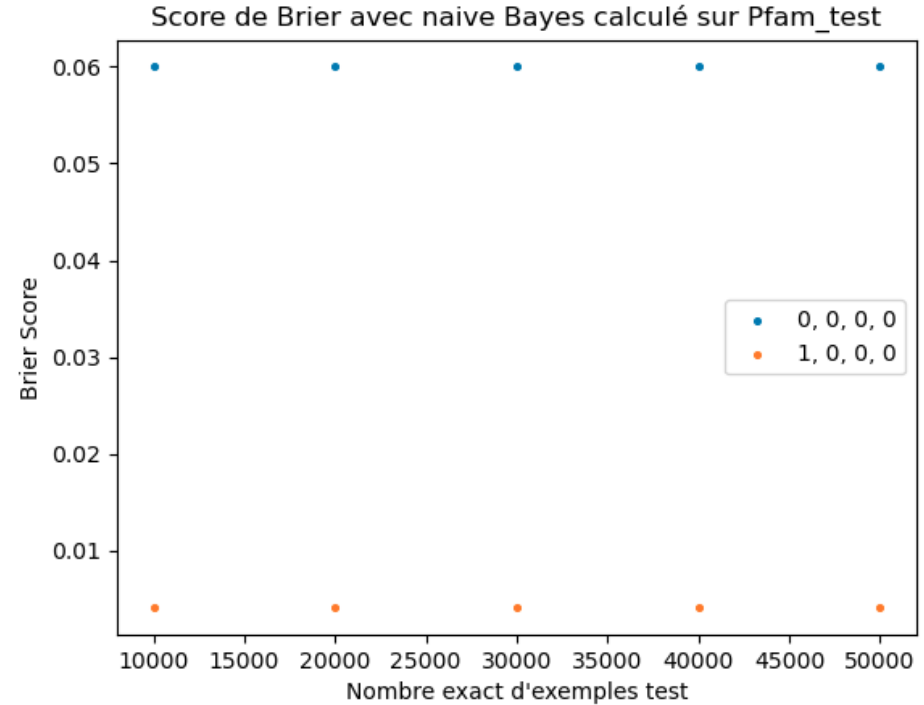
9 seeds identités, 1 seed décalé (cas décalé dans train)

CAS 3

Heatmap of the conditional probability matrix computed on Pfam_train



0-0- ... -0-0,1-0,8-0,1-0-0 ...0



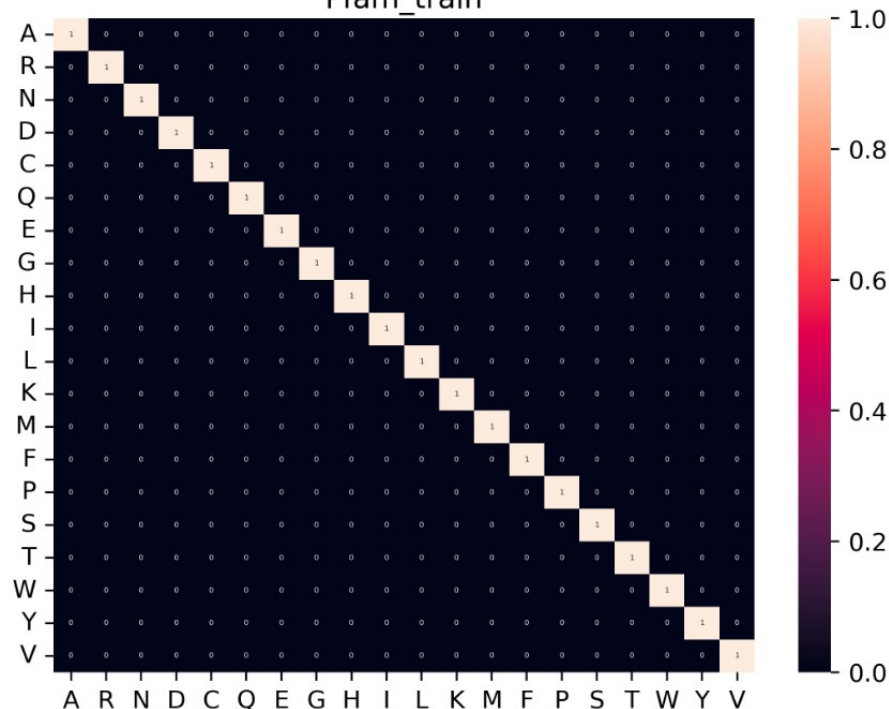
(0,0,0,0) : [0.059999999999999378, 0.059999999999996961, 0.059999999999996155, 0.0599999999999957525, 0.059999999999995511]

(1,0,0,0) : [0.004068480393967478, 0.004067730587873158, 0.004066591993432518, 0.0040691885441681355, 0.004067813899663722]

9 seeds identités, 1 seed décalé
(cas décalé pas dans train)

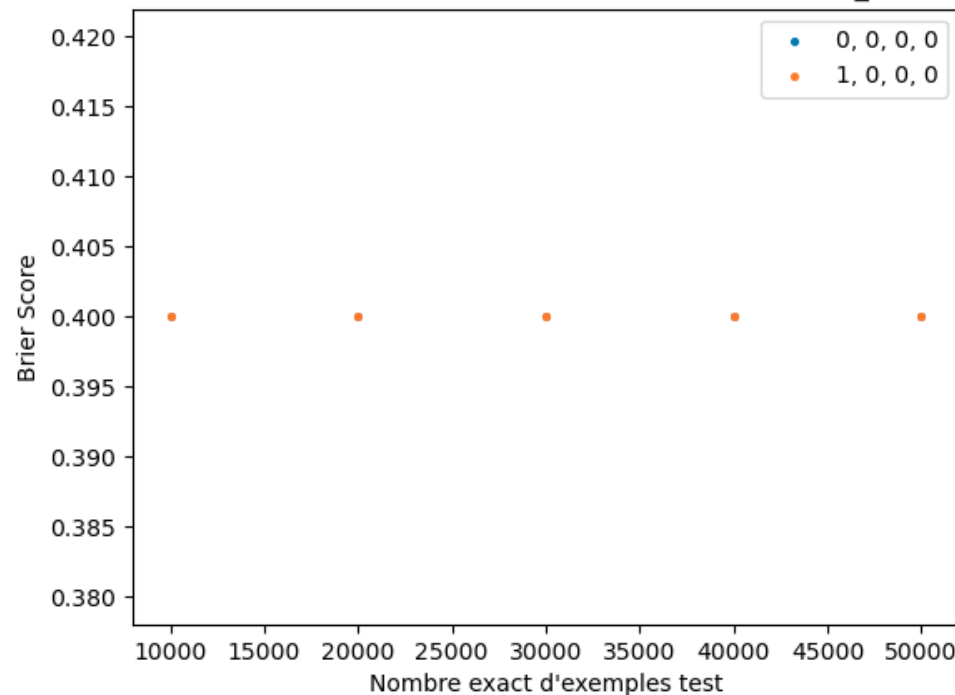
CAS 4

Heatmap of the conditional probability matrix computed on Pfam_train



0-0- ... 0-0-1-0-0 ...0

Score de Brier avec naive Bayes calculé sur Pfam_test



(0,0,0,0) : [0.4, 0.4, 0.4, 0.4, 0.4]

(1,0,0,0) : [0.4, 0.4, 0.4, 0.4, 0.4]

ANNEXES

Prob affichage/ veille

brier_naive_bayes_v1.py - MNHN_EvolProt - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- ▼ MNHN_EVOLPROT
 - selection_ex... M
 - > cluster
 - > dossier_test_score...
 - ▼ localNeighbour
 - > __pycache__
 - localNeighbourfo...
 - scriptCube_1DK.py
 - scriptCube_1DP.py
 - scriptCube_1GK.py
 - scriptCube_1GP.py
 - scriptCubeRef.py
 - scriptCubeRef.sh
 - ▼ test_prelimin...
 - brier_naive... M
 - draft_seed_select...
 - main_brier_no_co...
 - > treatment
 - > utils
 - main_blosum.py M
 - main_brier_no_con...
 - main_data_treatm...
 - main_local_neigh...
 - main_script_cube...
 - pfam_residu_desc.py
 - README.md
 - seed_non_informat...
 - .gitignore
 - Résultats_surpr... U
 - > OUTLINE
 - > TIMELINE

MNHN > test_preliminaire_voisinage_local > brier_naive_bayes_v1.py > naive_bayes_brier

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4
5 import sys
6 from pathlib import Path
7 file = Path(__file__).resolve()
8 package_root_directory_MNHN = file.parents[2]
9 root_path = file.parents[3]
10 sys.path.append(str(package_root_directory_MNHN))
11
12 import MNHN.brierNeighbour.selection_example as selection_example
13 from MNHN.utils.timer import Timer
14
15 def cube_loader(max_relative_distance, k_or_p, l_or_r, path_cube_folder):
16     """
17     max_relative_distance: indice le plus lointain dans quart de fenetre
18     k_or_p: séquence d'origine (k) ou de destination (p)
19     l_or_r: voisinage à gauche ou à droite
20     """
21     # initialisation de la liste des cubes pour un quart de fenetre contextuelle
22     list_cube_quarter_window = []
23
24     for i in range(1, max_relative_distance + 1):
25         if l_or_r == "l":
26             path_cube = f"{path_cube_folder}/proba_cond_{i},{k_or_p}.npz"
27             list_cube_quarter_window.append(np.load(path_cube, allow_pickle='TRUE').
28
29         if l_or_r == "r": # je sais qu'un else marche mais if pour le moment
30             path_cube = f"{path_cube_folder}/proba_cond_{i},{k_or_p}.npz" # à vér
31             list_cube_quarter_window.append(np.load(path_cube, allow_pickle='TRUE').
32
33     return list_cube_quarter_window
34
35
```

Ln 129, Col 50 Spaces: 4 UTF-8 LF Python 3.9.13 ('base': conda)

Temps clustering 99 % Pfam et Pfam en sortie de traitement

```
99.99490627546862, PF17407.5
---> time redundant: 2.03332 s
100.0, PF02713.17
---> time redundant: 0.02593 s
---> Compute and save non-redundant files: 59154.88105 s
nbre seed:                19 632.00
nbre seq:                  1 235 590.00
nbre position:             4 448 999.00
total character:           346 322 402.00
total character included:  192 394 396.00
mean len seq:              155.71
mean nbre seq:             62.94
---> Split data total in data A and data B: 2.01514 s
(base) pauline@abiboom:~/MNHN_EvolProt$
```