

Le rapport scientifique

01/10/18

Quelques remarques préliminaires

- Un rapport s'écrit au “nous” majestatif
- Le jupyter notebook doit se suffire à lui-même ! (+ éventuelles images)
- Le code est en python 3 et doit tourner. Code qui ne tourne pas = 0

The structure of a report

- Introduction
- Matériel & Méthodes
- Résultats
- Discussion (& Conclusion)

Introduction = QUOI ?

Quel est le sujet ?

Quelles connaissances
sont nécessaires pour
comprendre ?

Quel est le
problème abordé ?

Quels sont brièvement
les moyens utilisés
pour le résoudre ?



Matériel & Méthodes = **COMMENT** ?

Datasets utilisés

Formules mathématiques
ou pseudocode/algorithme

Fonction(s) (groupées) avec
explications

Proof of concept !



Résultats = QU'EST-CE QUE ?

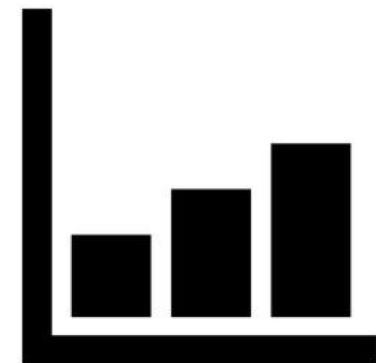
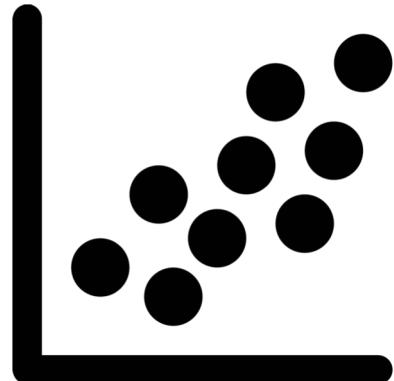
Utilisez le code sur les datasets

Qu'observez-vous ?

Différences entre les datasets ?

Test avec d'autres paramètres ?

Comparaisons ?



Discussion = POURQUOI ?

Vous avez vu quelque chose,
mais d'où ça vient ?

Basez-vous sur ce que vous
savez, répondez aux
questions posées.

Hypothèses, pistes de
réflexion,...



Conclusion

Résumez.

Question → Méthode pour le
résoudre → Observation(s)
→ Réponse ou piste



The good, the bad & the ugly

Ou apprendre du passé pour construire un meilleur future ! (Et éviter que l'assistante ne nous assassine)

Good

Etape 2 : Remplissage des matrices S, V, W, BT (fill_s_bt_v_w(self, type))

Le remplissage des autres éléments des 3 matrices se fait selon le raisonnement suivant :

$$V(i,j) = \max \begin{cases} S(i-1,j) - I \\ V(i-1,j) - E \end{cases}$$

$$W(i,j) = \max \begin{cases} S(i,j-1) - I \\ W(i,j-1) - E \end{cases}$$

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + t(i,j) \\ V(i,j) \\ W(i,j) \end{cases}$$

L'alignement courant optimal est calculé en décidant quel scénario match/dismatch ou gap donne le score le plus élevé (d'où l'utilisation de la fonction max) à partir de l'alignement optimal précédent.

Pour le remplissage de $S(i, j)$ le choix fait par la fonction max permet de former l'alignement. Trois choix sont possibles:

1. $S(i-1, j-1) + t(i,j)$: Ce qui signifie aligner les 2 caractères des deux séquences
2. $V(i,j)$: Ce qui signifie ajouter un espace dans la deuxième séquence
3. $W(i,j)$: Ce qui signifie ajouter un espace dans la première séquence

Pour le remplissage de V et W les 2 choix correspondent aux 2 possibilités :

1. La valeur précédente n'était pas un gap
2. La valeur précédente était un gap

Pour faciliter l'étape 3 une matrice supplémentaire BT est remplie en même temps que la matrice S. Si le cas 1 de la fonction max est retenu un "D" est ajouté à $BT(i, j)$ car cela signifie que la valeur (i,j) à été obtenue grâce à la case qui se situe une unité en haut et une unité à droite donc bien dans sa diagonale, si le cas 2 de la fonction max est retenu un "T" pour "top" est ajouté à $BT(i, j)$ et si le cas 3 est retenu "L" pour "left" est ajouté à $BT(i, j)$. Ces lettres servent à indiquer le chemin retour à effectuer.

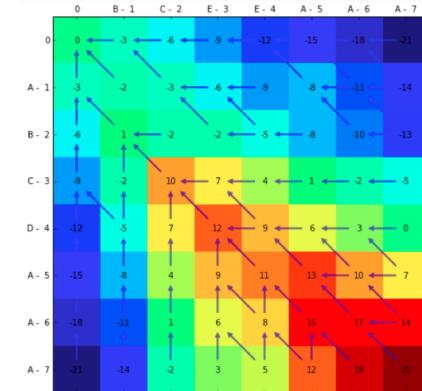
Etape 3 : Formation de l'alignement (global_align(self, i=-1, j=-1))

Pour former l'alignement il est nécessaire de remonter la matrice S (l'alignement est formé de droite à gauche). La remontée commence en dernière ligne dernière colonne et se termine en première ligne première colonne (d'indice 0). Grâce à la matrice BT construite à l'étape 2 qui indique les directions à suivre il est facile de retrouver les chemins possibles en utilisant un backtracking classique. A chaque étape il faut itérer sur toutes les directions possibles. A la fin de la fonction tous les meilleurs alignements ont été reformés. Le backtracking a lieu dans la fonction `global_align(self, i)`. La recursion s'arrête lorsque nous sommes arrivés en (0,0). C'est la fonction `global_align(self, i=-1, j=-1)` qui est chargée d'exécuter ce processus.

Un exemple

Le premier pas est de construire une matrice des scores. Les scores représentent la distance d'édition entre des parties des deux séquences.

Soient les 2 séquences "ABCDAAA" et "BCEEAAA". Voici la matrice des scores qu'on souhaite construire:



Séquence 1: no_meta -- THISLINE
Séquence 2: no_meta -- ISALIGNED
Pénalité I: 4
Pénalité E: 1

Score: 10
THIS-LI-NE-
--ISALIGNED

Matrice S:

	I	S	A	L	I	G	N	E	D	
I	0	-4	-5	-6	-7	-8	-9	-10	-11	-12
T	-4	-1	-3	-5	-7	-8	-9	-9	-11	-12
H	-5	-5	-2	-5	-7	-8	-9	-8	-9	-12
I	-6	-1	-5	-3	-3	-3	-7	-8	-9	-10
S	-7	-5	3	-1	-2	-3	-3	-5	-6	-7
L	-8	-5	1	2	3	0	-2	-3	-4	-5
I	-9	-4	-2	-2	4	7	3	2	1	0
N	-10	-8	-3	-3	0	3	7	9	5	4
E	-11	-9	-4	-4	-1	2	3	7	14	10

The score on the left is our result, whereas the result on the right is the one obtained from LALIGN

Sequences	1	2	3	4	5
1	NaN / NaN	-- / --	-- / --	-- / --	-- / --
2	99 / 109	NaN / NaN	-- / --	-- / --	-- / --
3	99 / 101	122 / 142	NaN / NaN	-- / --	-- / --
4	82 / 95	108 / 117	70 / 88	NaN / NaN	-- / --
5	122 / 125	98 / 112	95 / 106	76 / 92	NaN / NaN

Our implementation is not too far off from LALIGN's.

Bad...

- Pas de texte
- Pas d'explications ou juste descriptions
- Gros blocs de code
- ...

work hard now. it'll pay off later.



chibird