

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | TECHNIQUES DE RÉDUCTION DE DIMENSION : UMAP et TMAP | | | | |  | |
|  |  | | | | | | |  |
|  | | | | MEHDI DOUBIANI |  | | | |
|  | | | | 06/05/2020  — Projet 7 OC Développez une preuve de concept — |  | | | |
|  | | |  | | |  | | |

Contents

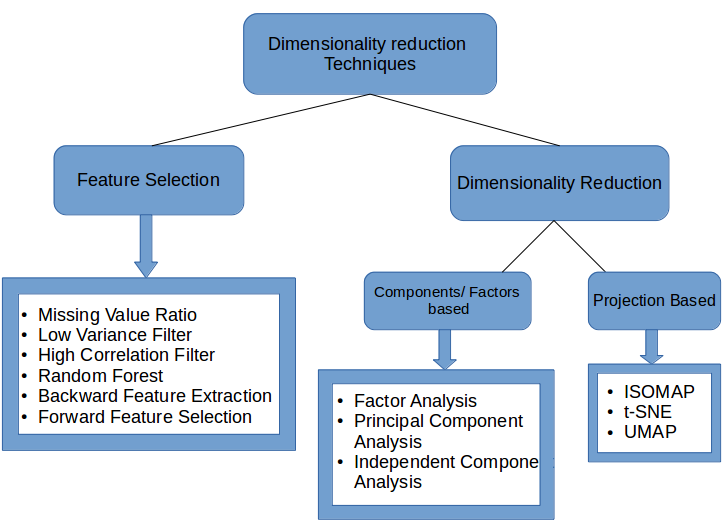
[TECHNIQUES DE RÉDUCTION DE DIMENSION : UMAP et TMAP 1](#_Toc40702556)

[Projet 7 OC Développez une preuve de concept 1](#_Toc40702557)

[1. Qu'est-ce que la réduction de la dimensionnalité ? 3](#_Toc40702558)

[2. Pourquoi la réduction de la dimensionnalité est-elle nécessaire ? 4](#_Toc40702559)

[3. ETAT DE L’ART : Résumé succinct des techniques de réductions des dimensions et quand les utiliser : 5](#_Toc40702560)

[ 5](#_Toc40702561)

[4. Comparaison PCA vs T-SNE vs UMAP dans l’absolu 6](#_Toc40702562)

[4.1. Analyse en composantes principales (ACP) 6](#_Toc40702563)

[4.2. t-Stochastic Neighbourhood Embedding (t-SNE) 6](#_Toc40702564)

[4.3. Uniform manifold approximation and projection (UMAP) 7](#_Toc40702565)

[5. Exemple de travail : Analyse comparative PCA vs TNE vs UMAP sur l’exemple de l’expression des ARN impliqués dans les cancers : 7](#_Toc40702566)

[6. Autres exemples : 10](#_Toc40702567)

[7. Comment fonctionne l'UMAP 11](#_Toc40702568)

[7.1. Comment (mal)interpréter l'UMAP 13](#_Toc40702569)

[8. Comment fonctionne TMAP 13](#_Toc40702570)

[9. Sources bibliographiques : 14](#_Toc40702571)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | TECHNIQUES DE RÉDUCTION DE DIMENSIONS | | |  | |
|  | |  |  |  |  | |
|  | Qu'est-ce que la réduction de la dimensionnalité ? Nous générons quotidiennement une quantité énorme de données. En fait, 90 % des données dans le monde ont été générées au cours des 3 ou 4 dernières années ! Les chiffres sont vraiment gigantesques. Vous trouverez ci-dessous quelques exemples du type de données collectées :   * Facebook recueille des données sur ce que vous aimez, partagez, postez, les endroits que vous visitez, les restaurants que vous aimez, etc. * Les Applications de votre smartphone recueillent de nombreuses informations personnelles vous concernant * Amazon collecte des données sur ce que vous achetez, visualisez, cliquez, etc. sur son site * Les Casinos gardent une trace de chaque mouvement de chaque client.   À mesure que la production et la collecte de données augmentent, il devient de plus en plus difficile de les visualiser et d'en tirer des informations utiles. L'une des méthodes les plus courantes de visualisation consiste à utiliser des graphiques. Supposons que nous ayons 2 variables, l'âge et la taille. Nous pouvons utiliser un diagramme de dispersion ou un graphique linéaire entre l'âge et la taille et visualiser facilement leur relation :    Imaginons maintenant des données où nous avons, disons, 100 variables (p=100). Dans ce données, nous pouvons avoir 100\*(100-1)/2 = 5000 tracés différents. Il ne serait pas très logique de visualiser chacune d'entre elles séparément, n'est-ce pas ? Dans les données où nous avons un grand nombre de variables, il est préférable de sélectionner un sous-ensemble de ces variables (p<<100) qui capture autant d'informations que l'ensemble original de variables.  Illustrons cela à l'aide d'un exemple simple. Considérons l'image ci-dessous :    Ici, nous avons les poids d'objets similaires en Kg (X1) et en livres (X2). Si nous utilisons ces deux variables, elles transmettront des informations similaires. Il serait donc logique de n'utiliser qu'une seule variable. Nous pouvons convertir les données de 2D (X1 et X2) en 1D (Y1) comme indiqué ci-dessous :    De même, nous pouvons réduire les p dimensions des données en un sous-ensemble de k dimensions (k<<n). C'est ce qu'on appelle la réduction de dimensionnalité. | | | | |  |

# Pourquoi la réduction de la dimensionnalité est-elle nécessaire ?

Voici quelques-uns des avantages de l'application de la réduction de la dimensionnalité à un ensemble de données :

L'espace nécessaire pour stocker les données est réduit au fur et à mesure que le nombre de dimensions diminue.

Moins de dimensions entrainent moins de temps de calcul/entrainement de modèles

Certains algorithmes ne fonctionnent pas bien lorsque nous avons de grandes dimensions. Il faut donc réduire ces dimensions pour que l'algorithme soit effectif.

Elle prend en charge la multicolinéarité en supprimant les caractéristiques redondantes. Par exemple, vous avez deux variables : "temps passé sur le tapis roulant en minutes" et "calories brulées". Ces variables sont fortement corrélées, car plus vous passez de temps à courir sur un tapis roulant, plus vous brulez de calories. Il est donc inutile de stocker les deux variables, car une seule d'entre elles vous informe sur ce que vous voulez étudier.

Elle aide à visualiser les données. Comme nous l'avons vu précédemment, il est très difficile de visualiser des données dans des dimensions plus élevées, de sorte que réduire notre espace à deux ou trois dimensions peut nous permettre de tracer et d'observer des modèles plus clairement.

# ETAT DE L’ART : Résumé succinct des techniques de réductions des dimensions et quand les utiliser :

Il est important de comprendre où vous pouvez, et devriez, utiliser une certaine technique car elle permet d'économiser du temps, des efforts et de la puissance de calcul.

Ratio de valeur manquante (**Missing Value Ratio)**: Si l'ensemble de données comporte trop de valeurs manquantes, nous utilisons cette approche pour réduire le nombre de variables. Nous pouvons supprimer les variables comportant un grand nombre de valeurs manquantes

Filtre à faible variance (**Low Variance filter)**: Nous appliquons cette approche pour identifier et supprimer les variables constantes de l'ensemble de données. La variable cible n'est pas indument affectée par les variables à faible variance, et donc ces variables peuvent être supprimées sans risque

Filtre à haute corrélation (**High Correlation filter**): Une paire de variables ayant une corrélation élevée augmente la multicolinéarité dans l'ensemble de données. Nous pouvons donc utiliser cette technique pour trouver des caractéristiques fortement corrélées et les supprimer en conséquence

Random Forest : C'est l'une des techniques les plus utilisées qui nous indique l'importance de chaque élément présent dans le jeu de données. Nous pouvons déterminer l'importance de chaque caractéristique et conserver le plus grand nombre de caractéristiques, ce qui entraîne une réduction de la dimension.

Les techniques d'élimination de variables non significatives en amont et de sélection de variables significatives en aval prennent beaucoup de temps de calcul et sont donc généralement utilisées pour des ensembles de données plus petits.

Analyse des facteurs (Factor Analysis): Cette technique est la mieux adaptée aux situations où nous avons un ensemble de variables fortement corrélées. Elle divise les variables en fonction de leur corrélation en différents groupes, et représente chaque groupe avec un facteur

Analyse en composantes principales (PCA): C'est l'une des techniques les plus utilisées pour traiter les données linéaires. Elle divise les données en un ensemble de composantes qui tentent d'expliquer autant de variance que possible

Analyse en composantes indépendantes (ICA): Nous pouvons utiliser l'ICA pour transformer les données en composantes indépendantes qui décrivent les données en utilisant un nombre réduit de variables.

ISOMAP : Nous utilisons cette technique lorsque les données sont fortement non linéaires

t-SNE : cette technique fonctionne également bien lorsque les données sont fortement non linéaires. Elle fonctionne aussi très bien pour les visualisations

LargeVis : une technique qui construit d'abord un graphique du voisin le plus proche K approximé avec précision à partir des données, puis dispose le graphique dans l'espace à faible dimension. Par rapport à t-SNE, LargeVis réduit considérablement le coût de calcul de l'étape de construction du graphique et utilise un modèle probabiliste de principe pour l'étape de visualisation, dont l'objectif peut être optimisé efficacement par une descente de gradient stochastique asynchrone avec une complexité temporelle linéaire.

UMAP : Cette technique fonctionne bien pour les données de haute dimension. Sa durée d'exécution est plus courte que celle du t-SNE

## 

# Comparaison PCA vs T-SNE vs UMAP dans l’absolu

## Analyse en composantes principales (ACP)

Considérons une matrice d'ensemble de données MxN avec M variables et N observations. Chaque observation de notre ensemble de données est un vecteur dans un espace vectoriel à M dimensions, couvert par une base orthonormée. Nous savons qu'il y a une redondance inhérente parmi les variables et que toutes les M dimensions ne sont pas importantes pour comprendre la dynamique de l'ensemble de données. L'ACP pose une question : Y a-t-il une autre base, qui est une combinaison linéaire de la base originale, qui exprime le mieux notre ensemble de données ?

L'ACP effectue une transformation orthogonale de l'ensemble de données original pour créer un ensemble de nouvelles variables ou composantes principales non corrélées. Ces composantes principales sont des combinaisons linéaires de variables dans l'ensemble de données d'origine. La transformation est définie de telle sorte que les composantes principales sont classées par ordre décroissant de variance. Ainsi, la première composante principale correspond à la variance la plus élevée possible. L'idée est d'éliminer les composantes principales ayant la variance la plus faible, et de réduire efficacement les dimensions de l'ensemble de données sans grande perte d'informations.

**Pour**

Hautement interprétable, peu coûteux en temps de calcul

**Contre**

Les données sur les séquences d'ARN sont rares en raison des non expressions (les gènes faiblement exprimés ne sont pas enregistrés), ce qui signifie qu'il y a 60 à 80 % de zéros dans la matrice de données. Il s'agit d'une structure hautement non linéaire, tandis que l'ACP est une technique de réduction dimensionnelle linéaire et donc jugée très inappropriée pour la visualisation des données. L'ACP n'est utilisée que pour sélectionner ~10 à 50 composants principaux qui peuvent être traités avec des applications en aval comme l'analyse des « clusters ».

## t-Stochastic Neighbourhood Embedding (t-SNE)

Le t-SNE est une technique de réduction de la dimensionnalité non linéaire basée sur un graphique. Elle projette des données à haute dimension sur des composants en 2D ou 3D et constitue une méthode de visualisation de données très populaire. En bref, l'algorithme crée d'abord une distribution de probabilité gaussienne qui définit les relations entre les points dans l'espace à haute dimension. Il utilise ensuite une distribution t de Student pour recréer la distribution de probabilité dans l'espace à faible dimension. Les données (embeddings) dans l'espace à faible dimension sont optimisés en utilisant la descente de gradient.

**Pour**

**Ensembles de données non linéaires :** t-SNE capture avec puissance la non-linéarité des ensembles de données de grandes dimensions et est capable de conserver les structures locales de petites dimensions. Il s'agit d'une amélioration considérable par rapport à l'ACP. Par exemple, la figure suivante montre une structure dites de rouleau suisse. La proximité entre les deux points mis en évidence, telle que calculée par l'ACP, sera une ligne droite joignant les deux points, alors que t-SNE, en théorie, devrait être capable de résoudre la non-linéarité du manifold en le déballant de manière appropriée.

**Méthode de pointe :** L'algorithme est conçu pour préserver les structures locales, ainsi les points proches dans l'espace à haute dimension finissent plus près dans les données à basse dimension. t-SNE a été utilisé comme une méthode de référence pour la visualisation des données de la séquence d'ARN.

**Contre**

**Aléatoire :** t-SNE est une méthode stochastique, elle implique une initialisation aléatoire et ne produit pas de résultats similaires lors de passages successifs. Bien que cela puisse être troublant pour les utilisateurs, il est important de noter que les sous-structures locales sont préservées. Par conséquent, tant que vous autorisez un nombre suffisant d'itérations pour permettre la convergence, la sortie globale ne change pas beaucoup. De plus, vous pouvez toujours définir une graine pour vous assurer que le résultat est le même à chaque fois que vous exécutez l'algorithme.

Par exemple, les images suivantes sont les sorties t-SNE d'un programme exécuté deux fois avec les mêmes paramètres d'entrée (sans définir une graine aléatoire). Comme vous pouvez le voir, les grappes changent de position mais les deux sorties produisent un résultat globalement similaire, c'est-à-dire 5 « clusters ».

**La structure globale n'est pas conservée :** De la manière dont t-SNE fonctionne, il lui est impossible de préserver la structure globale tout en effectuant une réduction de dimension. Ceci est probablement lié à l'optimisation de la divergence Kullback-Liebler.

**Visualisation des données uniquement :** t-SNE intègre les points de données sur 2 ou 3 dimensions maximum uniquement. Contrairement à l'ACP, il ne produit pas de 10 à 50 composants pouvant être exploités par un algorithme de clustering. t-SNE, en tant que technique de réduction de la dimensionnalité, se limite donc à l'exploration ou à la visualisation des données.

**Coûteux en termes de calcul :** Pour des ensembles de données volumineux, l'algorithme prend beaucoup de temps à s’exécuter. Il est courant de réduire d'abord les dimensions par ACP, puis d'effectuer des t-SNE sur les quelques composants principaux les plus importants.

Bien qu'une mise en œuvre plus rapide, appelée FItsne, ait été conçue, elle souffre encore des autres inconvénients de t-SNE.

## Uniform manifold approximation and projection (UMAP)

L'UMAP est une technique de réduction dimensionnelle relativement nouvelle introduite par McInnes et al en 2018. L'algorithme est basé sur un graphe et est principalement similaire à l'algorithme t-SNE où il construit une représentation graphique à haute dimension des données, puis optimise un graphe à basse dimension pour être aussi structurellement similaire que possible.

**Pour**

**Ensembles de cas non linéaires :** L'UMAP est une technique de réduction de la dimension d'apprentissage multiple et permet ainsi de saisir la non-linéarité des ensembles de données du monde réel. Elle est comparable à t-SNE en termes de visualisation des données.

**Efficacité de calcul :** Les améliorations mathématiques apportées à UMAP permettent d'obtenir des performances de calcul supérieures à celles de t-SNE (et de l'algorithme improvisé FItsne).

Préservation de la structure globale : Par rapport au t-SNE, UMAP offre une meilleure préservation de la structure globale des données. En réglant les hyperparamètres n\_neighbours et min\_dist, les utilisateurs peuvent contrôler efficacement l'équilibre entre les structures locales et globales.

**Applications en aval :** Contrairement à t-SNE, l'UMAP n'a pas de restrictions de calcul sur les dimensions en entrée et peut être utilisé comme une étape de prétraitement efficace pour augmenter les performances des algorithmes de clustering basés sur la densité.

**Contre**

**Manque d'interprétabilité :** Contrairement à l'ACP, dont les principales composantes sont les directions de la plus grande variance des données sources, les données de dimension inférieure de l'UMAP ne sont pas facilement interprétables.

**Détection de bruit parasite :** L'une des hypothèses fondamentales de l'UMAP est qu'il existe une structure multiple dans les données. De ce fait, UMAP peut avoir tendance à trouver une structure multiple dans le bruit d'un ensemble de données. L'UMAP est plus robuste avec des ensembles de données plus importants, car la quantité de structure évidente dans le bruit a tendance à diminuer dans les ensembles de données plus importants.

**Conservation de la structure globale :** Bien que l'UMAP préserve davantage la structure globale que d'autres techniques comme t-SNE, il se préoccupe principalement de représenter avec précision la structure locale. Si la structure globale est d'un intérêt primordial, alors l'UMAP n'est peut-être pas le meilleur choix pour la réduction des dimensions.

# Exemple de travail : Analyse comparative PCA vs TNE vs UMAP sur l’exemple de l’expression des ARN impliqués dans les cancers :

**Fichier : P7\_02\_Cancer\_Data\_Code.ipynb**

Les abréviations des échantillons PRAD, LUAD, BRCA, KIRC et COAD correspondent à des types de cancers différents et donc des tissus différents :

-PRAD : Prostate Adenocarcinoma

- LUAD : Lung Adenocarcinoma

- BRCA : Breast Invasive Carcinoma

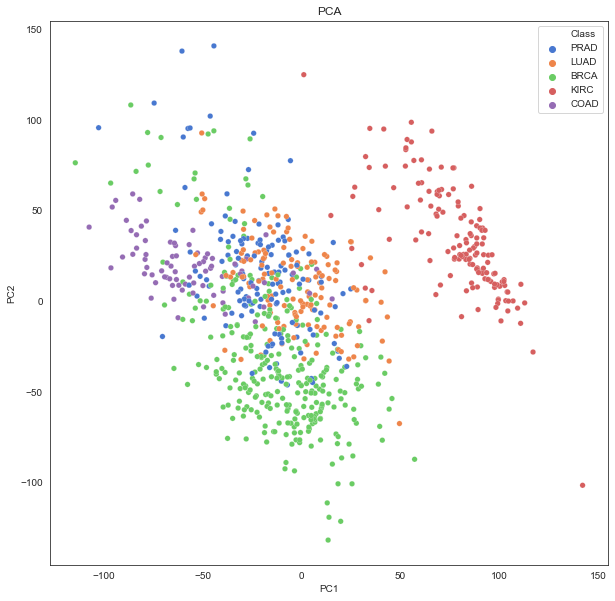
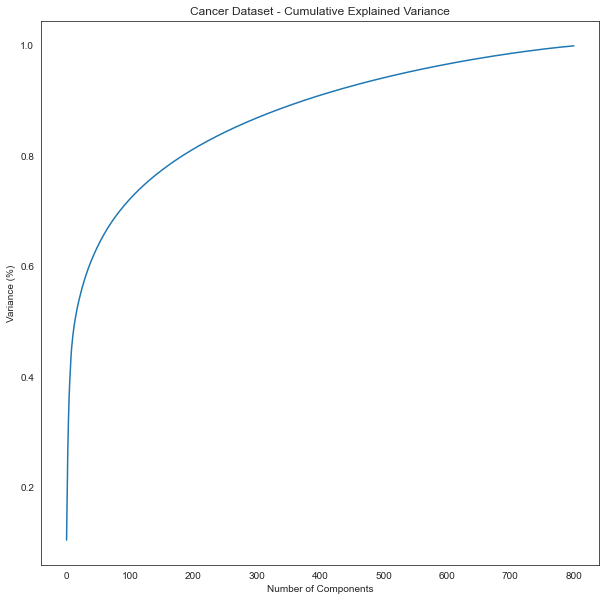
- KIRC : Kidney Renal Clear Cell Carcinoma

- COAD : Colon Adenocarcinoma

Après avoir passé en revue les avantages et les inconvénients de chaque méthode au point 4, évaluons-les sur la base d'un ensemble de données du monde réel. Je reconnais qu'un seul ensemble de données ne suffit pas, et que plusieurs ensembles de données de taille et de complexité différentes devraient être utilisés pour une évaluation complète

**ACP :**

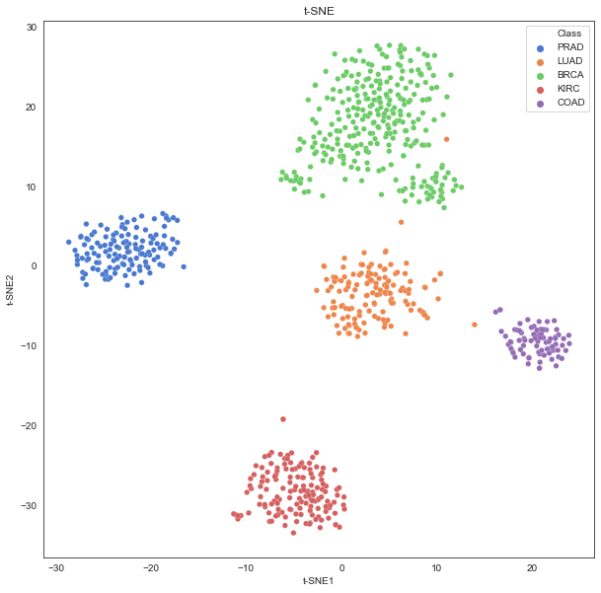
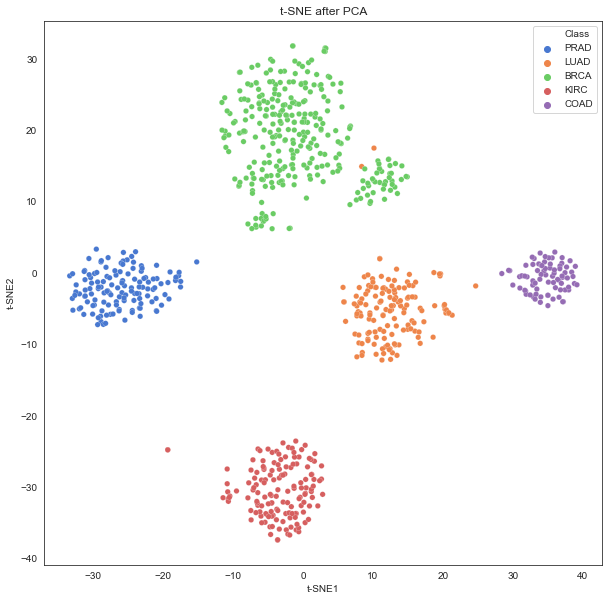
J'ai choisi un ensemble de données qui contient de véritables informations de regroupement, qui peuvent être utilisées comme point de comparaison dans l'analyse comparative. Il s'agit d'une aggrégation de l'expression des gènes de différents types de tissus cancéreux, maintenue par le projet d'analyse pancancéreuse de l'atlas du génome du cancer. Les échantillons (instances) sont stockés par ligne. Les variables (attributs) de chaque échantillon sont les niveaux d'expression génétique ARN-Seq mesurés par la plateforme illumina HiSeq. Un nom fictif (gene\_XX) est donné à chaque attribut. Le lien pour le téléchargement se trouve sur kaggle (Note : il ne s'agit pas d'un ensemble de données de type ARN-Seq sur une seule cellule, mais l'idée reste la même et sert notre objectif d'analyse comparative des techniques de réduction de dimension).



L'ACP est clairement sous-optimale pour retranscrire l'hétérogénéité des données. Bien qu'il existe plusieurs autres techniques de réduction de la dimensionnalité basées sur l'approche de factorisation matricielle, elles peuvent fonctionner mieux que l'ACP ; elles sont en général plus efficaces pour préserver la structure globale et mettent moins l'accent sur la préservation de la structure locale des données. Pour comprendre la dynamique des ensembles de données biologiques complexes, il est important de préserver la structure locale en conservant les connexions entre les points voisins. C'est là que les méthodes basées sur les graphiques de voisinage comme t-SNE et l'UMAP surpassent les méthodes basées sur la factorisation matricielle.

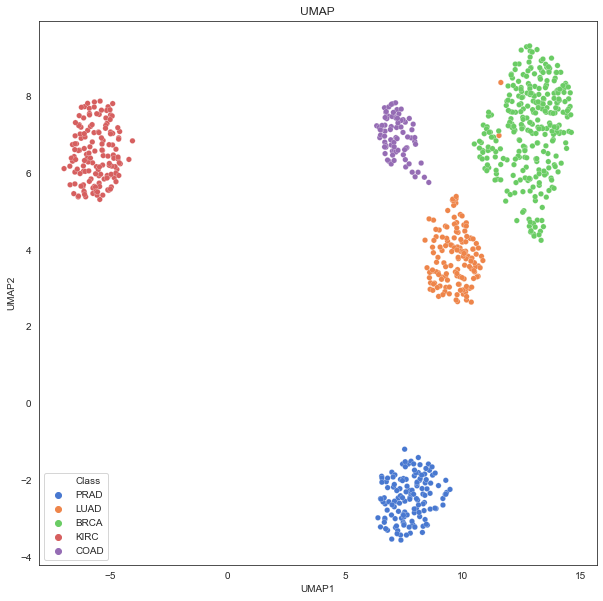
L'ACP est cependant encore largement utilisée comme étape de prétraitement pour les algorithmes de regroupement. Par exemple, dans notre ensemble de données sur le cancer, le graphique de la variance cumulative expliquée montre que les 100 premières composantes principales expliquent ~80 % de la variance des données. L'application d'algorithmes de mise en grappes (clustering) sur ces 100 CP accélérerait leurs performances sans trop compromettre la perte d'informations.

**t-SNE :**



Le t-SNE a fait un bien meilleur travail pour retranscrire les regroupements en grappes (clusters). Seuls 3 points de données du cluster LUAD (orange) sont incorrectement assignés comme BRCA et COAD. Le résultat est visuellement attrayant, pas étonnant qu'il ait été utilisé comme une technique de référence pour l'analyse des cellules individuelles au cours de la dernière décennie. Compte tenu des limites de t-SNE décrites plus haut, il s'agit d'une technique qui mérite d'être explorée avec un réglage approprié des paramètres.

**UMAP :**



L'UMAP a donné une résolution tout aussi bonne, voire meilleure, que t-SNE. Le fait qu'elle préserve également la structure globale dans une certaine mesure et qu'elle fonctionne beaucoup plus rapidement que t-SNE la rend très attrayante et il ne sera pas surprenant de voir les chercheurs se tourner davantage vers l'UMAP dans un avenir proche.

# Autres exemples :

**Fichier : P7\_02\_UmapPyCaret\_Autre\_Exemple.ipynb**

**Kiva dataset :** NLP Classification

Kiva Microfunds est une organisation à but non lucratif qui permet aux particuliers de prêter de l'argent à des entrepreneurs et des étudiants à faible revenu dans le monde entier. Depuis ses débuts en 2005, Kiva a financé des millions de prêts avec un taux de remboursement d'environ 98 %. Chez Kiva, chaque demande de prêt comprend à la fois des informations démographiques traditionnelles sur l'emprunteur, telles que le sexe et le lieu, ainsi qu'une histoire personnelle. Le texte donné dans l'histoire personnelle permet de comprendre l'ensemble des données et la structure sémantique cachée dans le texte. L'ensemble de données contient 6 818 échantillons. Vous trouverez ci-dessous une brève description des caractéristiques :

pays : pays de l'emprunteur

en : Histoire personnelle de l'emprunteur lors de la demande de prêt

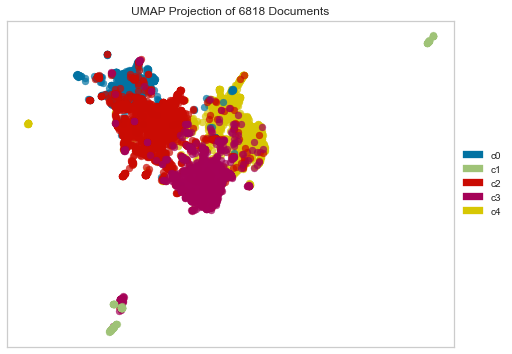
genre : sexe (M=homme, F=femme)

montant\_du\_prêt : Montant du prêt approuvé et déboursé

le non-paiement : Type de prêteur (Prêteur = utilisateur personnel enregistré sur le site web de Kiva, Partenaire = institution de microfinance qui travaille avec Kiva pour trouver et financer des prêts)

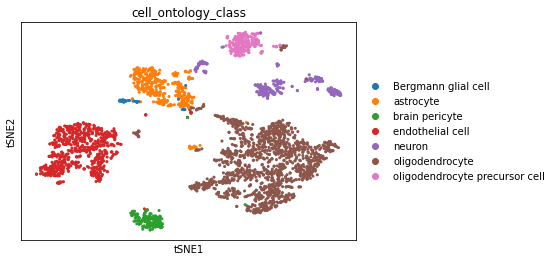
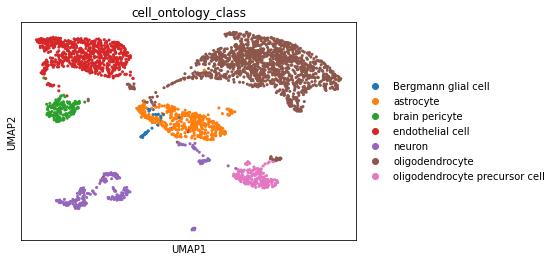
secteur : secteur de l'emprunteur

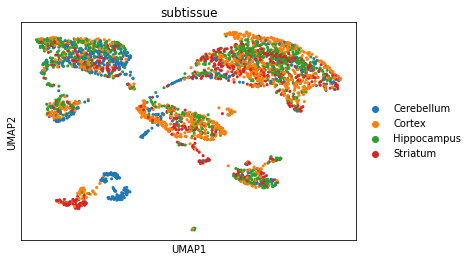
statut : statut du prêt (1-défaut, 0-remboursement)



On remarque ici que la visualisation est moins bonne que dans l’exemple biologique car les profils de personnes demandant des micros prêts est très similaire.

**Fichier : P7\_02\_Tabula\_Muris-Dimension\_Reduction\_Code\_Autre\_Exemple.ipynb**





Dans cet exemple, on visualise le niveau d’expression des gènes dans chaque type de cellules pour aider les chercheurs à comprendre comment s’expriment les protéines par types de tissus.

On remarque encore une fois que la méthode t-SNE offre une visualisation satisfaisante mais de par le temps de calcul la méthode UMAP est à privilégier.

# Comment fonctionne l'UMAP

L'algorithme UMAP est basé sur des concepts mathématiques lourds, mais nous allons ici essayer de le décomposer en étapes simples.

* Tout d'abord, nous regroupons les observations en groupes de k observations. Par défaut, la distance euclidienne est utilisée, mais vous pouvez utiliser n'importe quelle mesure de distance à la place.
* Les observations qui se trouvent à l'intérieur d'un groupe sont connectées, alors considérons cela comme un "graphe".
* Ces "graphiques" ont des bords pondérés, donc ces "graphiques" ne sont pas des composants connectés isolés, ils sont en quelque sorte connectés. Le poids de leurs bords fait que chaque observation appartient à un groupe de manière floue, plutôt que d'appartenir à un seul groupe de manière binaire.
* Tout ce temps, nous avons travaillé en haute dimension : jusqu'à présent, nous avons calculé une façon de représenter la structure de nos données dans la dimension haute. Mais le but était de la représenter dans une dimension inférieure ! Que devrions-nous faire alors ?
* N'oubliez pas que nos "graphiques" ont des bords pondérés. Ce que nous voulons faire, c'est calculer ces poids dans la dimension inférieure, mais nous voulons que ces poids restent aussi proches que possible de ceux de la dimension supérieure. Nous pouvons résoudre ce problème en tant que problème d'optimisation. Pour résumer, cette dernière étape, ce que nous voulons, c'est une représentation de bas niveau qui ressemble à la représentation de haut niveau (l'original).

Premièrement, il est considérablement plus rapide que t-SNE, où le temps nécessaire à son exécution augmente moins que le carré du nombre de données dans l'ensemble de données. Pour mettre les choses en perspective, un ensemble de données qui pourrait prendre des heures à calculer dans le cas de t-SNE prendra quelques minutes à l'UMAP.

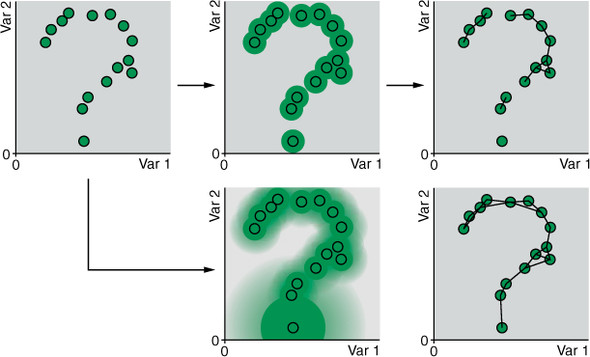
Le deuxième avantage (et le principal, à mes yeux) est que UMAP est un algorithme déterministe. En d'autres termes, avec la même entrée, il donnera toujours la même sortie. Cela signifie que, contrairement au t-SNE, nous pouvons projeter de nouvelles données sur la représentation à plus faible dimension, ce qui nous permet d'incorporer UMAP dans nos pipelines d'apprentissage machine.

Le troisième avantage est que l'UMAP préserve à la fois la structure locale et globale. Concrètement, cela signifie que non seulement nous pouvons interpréter deux données proches l'un de l'autre dans les dimensions inférieures comme étant similaires dans les dimensions supérieures, mais nous pouvons également interpréter deux groupes de données proches l'un de l'autre comme étant plus similaires dans les dimensions supérieures.

Comment fonctionne donc l'UMAP ? Eh bien, UMAP suppose que les données sont distribuées le long d’un manifold. Un manifold est une forme géométrique lisse à n dimensions où, pour chaque point de ce manifold, il existe une petite zone autour de ce point qui ressemble à un plan plat à deux dimensions. Si cela n'a pas de sens pour vous, considérez que le monde est une multiplicité tridimensionnelle, dont chaque partie peut être représentée par une carte plane littéralement appelée plan. UMAP recherche une surface, ou un espace à plusieurs dimensions, le long de laquelle les données sont réparties. Les distances entre les données le long du manifold peuvent alors être calculées, et une représentation en plus petites dimensions des données peut être optimisée de manière itérative pour reproduire ces distances.

Vous préférez une représentation visuelle ? Moi aussi.

Regardez la figure ci-dessous. J'ai dessiné un point d'interrogation sous la forme d'un manifold et j'ai semé au hasard 15 données autour du manifold sur 2 variables. Le travail de l'UMAP consiste à apprendre le point d'interrogation de la multiplicité afin qu'il puisse mesurer les distances entre les données le long de la multiplicité au lieu de la distance euclidienne ordinaire, comme le fait t-SNE. Il y parvient en recherchant une région autour de chaque donnée, pour une autre donnée. Lorsque ces régions encapsulent une autre donnée, les données sont reliées par un bord. C'est ce que j'ai fait à la première ligne de la figure ci-dessous, mais voyez-vous que le manifold est incomplet ? Il y a des lacunes dans mon point d'interrogation. C'est parce que les régions que j'ai recherchées autour de chaque donnée avaient le même rayon, et les données n'étaient pas uniformément réparties le long du manifold. Si les données avaient été espacés le long du point d'interrogation à intervalles réguliers, cette approche aurait fonctionné, à condition que j'aie sélectionné un rayon approprié pour les régions de recherche.



Comment UMAP apprend un manifold. UMAP étend une région de recherche autour de chaque donnée. Une forme naïve de cette méthode est présentée dans la rangée du haut, où le rayon de chaque région de recherche est le même. Lorsque des données dont les régions de recherche se chevauchent sont reliés par des bords, il y a des « trous » dans le manifold. Dans la rangée du bas, la région de recherche s'étend jusqu'au plus proche voisin, puis s'étend vers l'extérieur de manière floue, avec un rayon inversement proportionnel à la densité des données dans cette région. Il en résulte un manifold complet.

Les données du monde réel sont rarement distribuées de manière égale, et UMAP résout ce problème de deux manières. Premièrement, il étend chaque région de recherche pour chaque donnée jusqu'à ce qu'elle rencontre son voisin le plus proche. Cela garantit qu'il n'y a pas de données orphelines : alors qu'il peut y avoir plusieurs manifolds déconnectés dans un ensemble de données, chaque donnée doit se connecter à au moins une autre donnée. Deuxièmement, l'UMAP crée une région de recherche supplémentaire qui a un rayon plus grand dans les régions à faible densité et un rayon plus petit dans les régions à forte densité. Ces régions de recherche sont décrites comme floues, en ce sens que plus une donnée est éloignée du centre, plus la probabilité qu'il y ait un bord entre ces données est faible. Cela oblige à une distribution uniforme artificielle des données (et c'est de là que vient l'"uniforme" dans UMAP). Ce processus est représenté dans la ligne inférieure de la figure ci-dessus ; notez que nous obtenons maintenant une estimation plus complète de la multiplicité sous-jacente.

L'étape suivante consiste à placer les données sur un nouveau manifold dans (généralement) deux nouvelles dimensions. Ensuite, l'algorithme remanie itérativement ce nouveau manifold jusqu'à ce que les distances entre les données le long du manifold ressemblent aux distances entre les données le long du manifold original, à haute dimension. Cette étape est similaire à l'étape d'optimisation de t-SNE, sauf que l'UMAP minimise une fonction de perte différente appelée entropie croisée (cross-entropy) (alors que t-SNE minimise la divergence KL).

## Comment (mal)interpréter l'UMAP

Bien que l'UMAP offre un certain nombre d'avantages par rapport à t-SNE, elle n'est en aucun cas une solution miracle - et la lecture et la compréhension de ses résultats exigent une certaine prudence :

1. Les hyperparamètres sont très importants

Le choix de bonnes valeurs n'est pas facile et dépend à la fois des données et de vos objectifs (par exemple, le degré de précision de la projection). C'est là que la rapidité d'UMAP est un grand avantage - En exécutant UMAP plusieurs fois avec une variété d'hyperparamètres, vous pouvez avoir une meilleure idée de la façon dont la projection est affectée par ses paramètres.

2. La taille des clusters dans un tracé UMAP ne signifie rien

Tout comme dans t-SNE, la taille des groupes les uns par rapport aux autres est essentiellement dénuée de sens. C'est parce que l'UMAP utilise des notions locales de distance pour construire sa représentation graphique à haute dimension.

3. Les distances entre les clusters peuvent ne rien signifier

De même, les distances entre les grappes risquent d'être non significatives. S'il est vrai que les positions globales des clusters sont mieux préservées dans l'UMAP, les distances entre eux ne sont pas significatives. Encore une fois, cela est dû à l'utilisation des distances locales lors de la construction du graphique.

4. Le bruit aléatoire n'a pas toujours l'air aléatoire.

En particulier aux faibles valeurs de n\_neighbors, on peut observer des regroupements parasites.

5. Vous pouvez avoir besoin de plus d'un graphique

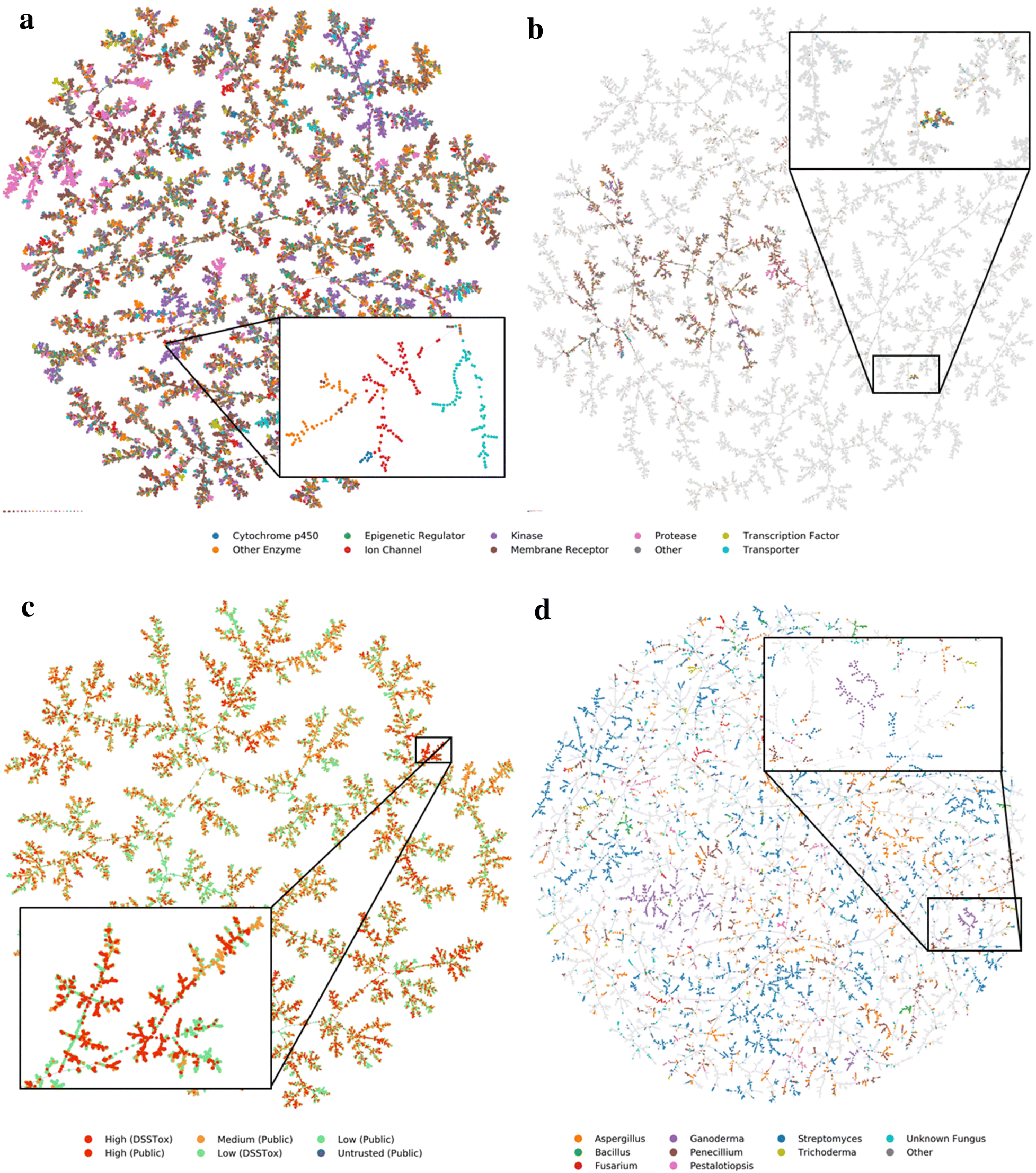
Comme l'algorithme UMAP est stochastique, différentes exécutions avec les mêmes hyperparamètres peuvent donner des résultats différents. De plus, le choix des hyperparamètres étant très important, il peut être très utile d'exécuter la projection plusieurs fois avec différents hyperparamètres.

# Comment fonctionne TMAP

TMAP est une méthode de visualisation de très grands ensembles de données à haute dimension permettant une grande interprétabilité des données en préservant et en visualisant des caractéristiques à la fois globales et locales. En utilisant TMAP en combinaison avec l'empreinte digitale MHFP6, nous pouvons visualiser des bases de données de millions de petites molécules organiques et les données de propriétés associées avec un haut degré de résolution, ce qui n'était pas possible avec les méthodes précédentes. TMAP est également bien adapté à la visualisation d'ensembles de données arbitraires tels que des images, du texte ou des données de séquences d'ARN, ce qui laisse entrevoir son utilité dans un large éventail de domaines, notamment la linguistique informatique ou la biologie.

Le TMAP excelle par sa faible utilisation de la mémoire et sa durée d'exécution, avec des performances supérieures à celles d'autres algorithmes de visualisation tels que le t-SNE, l'UMAP ou le PCA. En ajustant les paramètres disponibles et en tirant parti de la qualité de sortie et de l'utilisation de la mémoire, TMAP ne nécessite pas de matériel spécialisé pour des visualisations de haute qualité d'ensembles de données contenant des millions de points de données. Plus important encore, TMAP génère des visualisations avec une complexité temporelle sub-linéaire empirique de O(n exposant 0.931), permettant de visualiser des ensembles de données de haute dimension beaucoup plus importants que les méthodes précédentes.

Toutes les visualisations TMAP présentées, y compris les instructions d'installation et d'utilisation, sont disponibles sous forme de versions interactives en ligne (http://tmap.gdb.tools). Le code source de TMAP est disponible sur GitHub (https://github.com/reymond-group/tmap) et un paquet Python peut être obtenu en utilisant le gestionnaire de paquets conda.



# Sources bibliographiques :

UMAP

<https://arxiv.org/abs/1802.03426>   : Papier de recherche

<https://github.com/lmcinnes/umap>   : Implémentation Python

<https://umap-learn.readthedocs.io/en/latest/>   : Documentation

<https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668> : explication détaillée du fonctionnement de l’algorithme

<https://towardsdatascience.com/tsne-vs-umap-global-structure-4d8045acba17?source=user_profile---------1-----------------------> : Comparatif des deux techniques sur la préservation de la structure globale des données

<https://towardsdatascience.com/why-umap-is-superior-over-tsne-faa039c28e99> : pourquoi UMAP est meilleur que T-SNE

<https://sudonull.com/post/61683-Overview-of-the-new-UMAP-dimension-reduction-algorithm-Is-it-really-better-and-faster-than-t-SNE-New>

<https://arxiv.org/abs/1602.00370> : LargeVis

<https://umap-learn.readthedocs.io/en/latest/supervised.html>

<https://duhaime.s3.amazonaws.com/apps/umap-zoo/index.html>

<https://pair-code.github.io/understanding-umap/supplement.html>

<https://pair-code.github.io/understanding-umap/>

TMAP

<https://link.springer.com/article/10.1186/s13321-020-0416-x#Fig2>

<http://tmap.gdb.tools/>

<https://github.com/reymond-group/tmap>

Explications des réductions de dimensions :

<https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/>