

Implémentation de la méthode ‘Time Curves’ en rust avec une interface python

- **Encadrant** : Romain Giot romain.giot@u-bordeaux.fr
- **Mots clés** : Visualisation d’information, Réduction de dimension, rust, python

Résumé

Le but de ce projet est d’implémenter l’algorithme `Time curves`¹ en langage `rust`² et de proposer un binding `python` pour l’utiliser facilement dans des projets `python`. Cet algorithme est décrit dans un article scientifique, et une implémentation `java` de ses auteurs est disponible pour comparaison.

Travail à réaliser

- Développer le backend de la méthode dans le langage `rust` dans une bibliothèque dédiée. La modélisation de la bibliothèque doit permettre de facilement remplacer une partie de l’algorithme par une autre variante et les différentes constantes sont paramétrables et pas écrites en dur dans le code. Il faut donc bien identifier chacune des parties de l’algorithme et clairement les différencier dans la modélisation et il est facile de coder de nouvelles versions par la suite (par exemple en ayant différente variante de la partie qui projette les points en 2d) . Le backend génère les données nécessaires à la visualisation de la courbe, mais ne les affiche pas.
- Ajouter au backend des fonctions utilitaires pour faciliter son utilisation. Il faut trouver quelles fonctions sont pertinentes. Par exemple, la méthode nécessite une matrice de distance, mais peut-être que l’utilisateur veut juste fournir des points et demander à l’outil de calculer cette matrice.
- Créer un binding `python` qui appelle le code `rust` afin de pouvoir l’utiliser dans des projets `python`.
- Ajouter différents frontends de visualisation ou formats d’exports qui se basent uniquement sur les données générées par le backend (en rust, en vega ou html/js/css, en tikz, en tulip).

¹<https://aviz.fr/~bbach/timecurves/>

²<https://www.rust-lang.org/>

- Créer un outil en ligne de commande `rust` qui prend les données et les affiche avec un frontend
- Reproduire les exemples originaux pour valider l'implémentation
- Créer un outil graphique `rust` qui permet interactivement de tester l'impact de la configuration de l'algorithme sur les données choisies

Livrables

Dépôt `gitub.u-bordeaux.fr` avec :

- le code source du projet (les différents crates `rust` et leurs tests unitaires, les tests de validation)
- la documentation du projet (architecture, exemples d'utilisation)
- le rapport du projet

Bibliothèques d'intérêt

- `clap` pour gérer les arguments en ligne de commande
- `serde` pour la sérialisation des données
- `pyo3` pour l'interface avec `python`

Prérequis

- Bonnes compétences et gout en programmation (`rust` n'est pas un prérequis, mais ne sera pas simple à apprendre)
- Algèbre linéaire