

Compresse-moi si tu peux !

Université libre de Bruxelles – Faculté de science

Section : BA3 en science informatique

Rhabouqi Ismail, Hamdach Oumaima, Boutaleb Ikram, Boutaleb Mohamed, Qudaih Ahmed

Dans notre vie numérique quotidienne,¹ nous échangeons et stockons d'énormes quantités de données : documents, images, vidéos, musiques, données scientifiques, etc. Ces fichiers peuvent parfois être très volumineux et occuper beaucoup d'espace. Pour pouvoir stocker ces informations efficacement et



les transmettre rapidement sur Internet, on utilise une technique appelée **compression de données**.

Mais qu'est-ce que la compression ? La compression permet de réduire la taille des fichiers sans perdre leur contenu. Par exemple, lorsqu'un fichier contient beaucoup de répétitions, il est possible de trouver une manière plus courte de le représenter.

Dans notre projet, l'un des objectifs est d'étudier la compression de texte en explorant la notion d'information et les algorithmes de compression sans perte, qui conservent toutes les informations du fichier original. Cependant, tous les textes ne se compressent pas facilement, il existe des limites qu'il faut connaître pour déterminer si un texte est difficile à compressible ou non. Pour cela, nous utilisons une notion fondamentale "**entropie de Shannon**", introduite par le mathématicien Claude Shannon. L'entropie mesure la quantité moyenne d'information contenue dans un message. Elle permet d'estimer le taux de compression maximal possible pour un texte donné. Plus un texte est redondant (prévisible), moins il contient d'information, plus son entropie est faible et plus il peut être compressé.

Dans ce projet, nous nous intéressons à deux algorithmes de compression :

¹ Images générées par l'intelligence artificielle

- **L'algorithme de Huffman** : cette méthode attribue des codes binaires plus courts aux caractères fréquents et des codes plus longs aux caractères rares. L'algorithme calcule la fréquence des caractères, puis construit **l'arbre de Huffman** pour générer les codes binaires.
- **L'algorithme de LZW (Lempel-Ziv-Welch)** : il fonctionne en créant un dictionnaire de séquences des caractères rencontrées dans le texte. Lorsqu'une séquence se répète, elle est remplacée par un code plus court.

Dans la partie pratique de notre projet, nous avons développé une application permettant de compresser des fichiers texte, et nous pouvons choisir l'algorithme à utiliser et comparer les résultats pour déterminer lequel compresse le mieux. Nous utilisons des données variées, comme des textes en français, aussi des séquences d'ADN qui composées uniquement de quatre lettres (A, C, G, T). Les séquences génétiques peuvent être extrêmement longues et contiennent souvent des structures répétitives, ce qui en fait un excellent exemple de données compréhensibles.

Pour vulgariser ces concepts et les rendre faciles et compréhensibles à tous, nous proposons plusieurs activités :

- **Explications théoriques** : nous commençons par expliquer les notions théoriques (répétitions, notion d'information et entropie). Ensuite, nous proposons de petites séquences d'ADN et invitons les participants à deviner lesquelles se compressent mieux.
- **Simulation des algorithmes** : notre application comporte deux pages qui reproduisent chaque étape des algorithmes étudiés, soit à partir de mots prédéfinis, soit à partir de mots saisis par les participants.
- **Un jeu de construction d'arbre de Huffman** : Les participants choisissent au hasard une enveloppe contenant une séquence d'ADN simple, puis codent cette séquence en binaire selon l'algorithme de Huffman. La construction de l'arbre se fait avec des cartes représentant les lettres et leurs fréquences, étape par étape.
- Pour les curieux, **des fiches explicatives** détaillent plus la compression. De plus **un site web** permet aux visiteurs de compresser des fichiers texte en choisissant l'algorithme souhaité et propose un petit jeu de construction de l'arbre de Huffman pour s'amuser tout en apprenant.