| **Module** | **Responsabilités** |
| --- | --- |
| **config.py** | Paramètres généraux (taille de la fenêtre, couleurs, constantes, chemins de fichiers et configuration du logging) |
| **error\_correction.py** | Ajout et gestion des noeuds liés au code correcteur et correction des erreurs dans le graphe |
| **tran\_tran.py** | Transcription de l'ADN en ARNm et traduction de l'ARNm en protéine (conversion de codon en acide aminé, gestion du promoteur et du signal de terminaison) |
| **genetic\_code.py** | Définition du code génétique, ajout des bases, classification, complémentarité, motifs, processus, et gestion des segments 3-mers et de leur dégénérescence |
| **gene\_expression.py** | Simulation de l'expression génique en combinant transcription et traduction |
| **constants.py** | Définition des constantes de coûts, stabilité et erreurs pour la construction des noeuds et des interactions |
| **codon\_graph.py** | Construction du sous-graphe codon : création des noeuds codons, association codon → acide aminé, et ajout du chemin codonique |
| **encode\_decode.py** | Encodage d’un message en une séquence de bases et décodage d’un message à partir d’un chemin extrait des noeuds (conversion via la base 4) |
| **gene\_contraintes.py** | Validation des contraintes biologiques (présence du promoteur, signal de terminaison, longueur correcte, présence des noeuds obligatoires, etc.) |
| **visualization.py** | Visualisation du graphe : positionnement par couche, dessin des noeuds et arêtes, annotation des interactions, et mise en évidence d’un chemin optimisé |
| **optimisation.py** | Calcul du chemin optimal en utilisant plusieurs algorithmes (Bellman-Ford, Dijkstra, A\*) et évaluation des poids multi-critères en tenant compte de la simulation d'expression génétique |
| **init\_graph.py** | Initialisation et assemblage du graphe global en appelant les fonctions de création des différentes couches |
| **graph\_layers.py** | Définition et configuration des différentes couches du graphe (bases, motifs, segments, régulation) |
| **main.py** | Point d'entrée de l'application : configuration du logging, parsing des arguments, orchestration des modules (initialisation du graphe, encodage, simulation, optimisation, visualisation) |
| **requirements.txt** | Liste des dépendances et instructions d’installation |