**Ce dossier contient tous les scripts Java que j'ai écrits. Cette méta-donnée explique l'utilisation de chaque dossier et des classes.**

Ce que j'ai créé :

- Intervener : Ce dossier contient le code pour l'intervention de coupe en forceps. J'ai créé la classe ***`CepsGHAthinnerComposition.java`***, qui effectue la coupe en respectant toute la cible souhaitée, mais en cherchant à minimiser l'écart entre la composition réelle des espèces et la composition cible.

- GA: Ce dossier contient toutes les classes Java que j'ai modifiées à partir des fichiers de Michelle pour inclure la composition cible (le 4ᵉ contrôle). Cependant, en raison de la vitesse de la simulation, j'ai parallélisé les étapes de GA pour gagner du temps dans ***`LM\_paralle`***. Il existe également une version séquentielle, que vous pouvez tester dans ***`LM\_sequentielle`***. Le dossier `**michelle**` contient les scripts originaux de Michelle avant l'inclusion de la composition cible.

Compilation et exécution :

- Pour compiler, utilisez la commande ***`ant clean compile***` ou `**ant compile`**.

- Pour lancer la simulation sur Linux :

- ***LM\_paralle*** :

sh capsis.sh -p script forceps.myscripts.LM\_paralle.SimulationLoubnaManagementPa "fichier de commande”

- ***LM\_sequentielle*** :

sh capsis.sh -p script forceps.myscripts.LM\_sequentielle.SimulationLoubnaManagementSeq "fichier de commande"

***- michelle:***

sh capsis.sh -p script forceps.myscripts.michelle.SimulationMichelleManagement "fichier de commande"