

{rtaxrefdb} : que faire quand mon API préférée a été cyberattaquée ?

Maxime Jaunatre¹

Matthias Grenié²

Résumé (max 300 mots)

{rtaxref} [1] est un package R d'accès à l'API du référentiel taxonomique national, TAXREF, du Muséum National d'Histoire Naturelle, une étape essentielle pour le travail impliquant des noms d'espèce en écologie. Ce package est très utilisé pour standardiser les noms d'espèce par rapport à la référence donnée annuellement par TAXREF. À l'été 2026, le Muséum est cyberattaqué, rendant l'API hors-ligne.

Fort heureusement, le Muséum partage sur un site de secours les tableaux plats correspondants à la base de données. Nous avons donc développé un nouveau package {rtaxrefdb}, imitant les fonctionnalités de {rtaxref}. Le nouveau package mobilise des fichiers locaux en les intégrant dans une base de données SQL.

Cette transcription de fonctionnalités implique de réfléchir aux parallèles entre API et base de données (recherche exacte, correspondance approximative, parallélisation), et à terme, à l'intégration entre les deux packages existants. En outre, nous avons tenté d'appliquer les meilleurs principes de développement pour interagir avec une base de données SQL. Et tout cela, en ayant toujours en tête la facilité et la transparence d'utilisation par un public non averti.

Mots-clés : Package - API - SQL - Taxonomie - Ecologie - Ingénieur

Développement

Cette présentation se veut être un retour d'expérience sur la création du package {rtaxrefdb} à partir des fonctionnalités existantes du package {rtaxref}. Après une introduction sur les enjeux scientifiques du référentiel taxonomique TAXREF, j'expliquerai rapidement l'utilisation de {rtaxref} [1] et sa dépendance à l'accès à l'API de TAXREF. Au passage, j'évoquerai rapidement les meilleures pratiques de développement des packages API qui nécessitent de mobiliser des outils spécifiques que ce soit {vcr} pour enregistrer et rejouer des requêtes, {httr2} pour effectuer des requêtes ou bien {mockr} pour simuler la présence ou l'absence de ressources en ligne.

Puis, je présenterai la démarche adoptée pour développer le package {rtaxrefdb}. Je parlerai de notre ambition de développer rapidement un package fonctionnel en suivant des étapes simples et notre souci de reproduire des fonctionnalités équivalentes de {rtaxref} mais mobilisant des tableaux plats et des bases de données en SQL.

¹Univ. Grenoble Alpes, INRAE, LESSEM, 38000 Grenoble, France ; maxime.jaunatre@inrae.fr

²Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS,LECA, 38000 Grenoble, France ; matthias.grenie@univ-grenoble-alpes.fr

J'aborderai en particulier les questions de meilleures pratiques de développement pour les packages faisant des requêtes à des bases de données locales. Comme tout package qui cherche à adopter les meilleures pratiques de développement possibles, les packages interrogeant les bases de données locales se doivent d'avoir des tests unitaires pour valider leurs fonctionnements. Un outil essentiel pour cela est le package `{dittodb}` qui permet d'effectuer des requêtes sur une base de données simulée. Malgré la petite taille des données, l'optimisation des requêtes SQL est importante car TAXREF peut nécessiter un grand nombre de comparaisons de tables. Un point-clé, dans le cadre de l'utilisation du package, est la correspondance approximative, car le référentiel taxonomique est notamment mobilisé pour vérifier les noms des espèces dans des données et corriger de potentielles erreurs de saisie. Dans `{rtaxref}` c'est le serveur qui se chargeait d'effectuer cette correspondance, tandis que dans `{rtaxrefdb}` c'est directement la base PostgreSQL ou bien SQLite qui doit implémenter cette fonctionnalité. Pour être le plus facile d'utilisation, un package mobilisant une base de données doit pouvoir s'utiliser sur plusieurs architectures de bases de données (PostgreSQL et SQLite dans notre cas) tout en étant transparent pour les utilisateur·rice·s. Malgré des outils similaires, les spécificités de chaque architecture rendent le développement de package de base de données plus difficile.

Enfin, je mettrai en perspective les fonctions des packages `{rtaxref}` et `{rtaxrefdb}` avec les avantages et inconvénients de travailler plutôt avec une API ou bien avec une base de données locale. J'expliquerai, en outre, nos perspectives de travail quant à l'intégration des deux packages de manière transparente pour les utilisateur·rice·s finales.

Bibliographie

- [1] M. Grenié et H. Gruson, « `rtaxref`: An R Client for TAXREF the French Taxonomical Reference API ». 2025. [En ligne]. Disponible sur: <https://github.com/Rekyt/rtaxref>