

# Canapy

## Outil d'analyse et d'annotation de vocalisations animales

### Keywords

Machine Learning, Recurrent Neural Network (RNN), Reservoir Computing, Web development, User interface, Data analysis, MLOps

### Mots-Clés

Intelligence Artificielle, Réseau de neurones récurrent, réservoir-calcul, développement web, interface utilisateur, analyse de données, MLOps

### Encadrants

Xavier Hinaut, chercheur Inria

[xavier.hinaut@inria.fr](mailto:xavier.hinaut@inria.fr)

Nathan Trouvain, doctorant Inria (**questions et contact**)

[nathan.trouvain@inria.fr](mailto:nathan.trouvain@inria.fr)

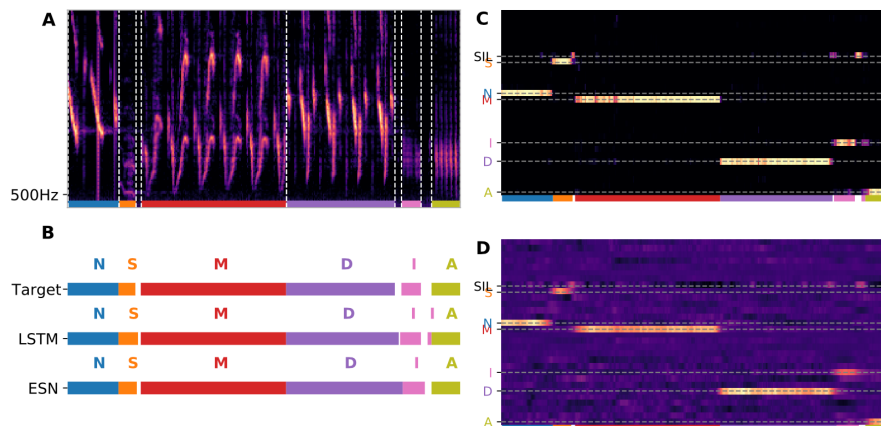
Arthur Leblois, chercheur IMN

[arthur.leblois@u-bordeaux.fr](mailto:arthur.leblois@u-bordeaux.fr)

### Équipe & Lieu du stage

Équipe Mnémosyne : Inria Bordeaux Sud-Ouest, LABRI & Institut des Maladies Neurodégénératives (Centre Broca Aquitaine, campus Carreire)

<https://team.inria.fr/mnemosyne>



Spectrogramme d'un chant de canari et reconnaissance par deux décodeurs (LSTM et ESN).

### Introduction et contexte scientifique

L'étude des oiseaux chanteurs est pertinente pour comprendre comment les enfants (humains) apprennent une langue, ou plus généralement pour comprendre comment notre cerveau apprend et traite des séquences sonores (e.g. musique, ...).

L'analyse de la structure des chants permettra de développer une meilleure compréhension et la création de nouveaux outils pour analyser des expériences faites en neurosciences.

Notamment pour l'analyse et le décodage de l'activité neuronale dans les régions cérébrales (chez l'oiseau) liés à l'apprentissage et la production du chant.

Cependant, l'analyse de ce type de données requiert un lent processus d'annotation, au cours duquel les enregistrements sonores sont triés et étiquetés à la main. De récents développements [1, 2] mettent à disposition des outils d'Apprentissage Artificiel (*Machine Learning ; ML*) pour automatiser une bonne partie de ce processus. Mais ces outils sont difficiles à maîtriser et à maintenir. Un premier prototype de service basé sur une interface web développée en Python (*panel, bokeh*) se propose d'offrir un outil simple permettant de mettre en place un *pipeline* d'annotation à l'aide d'une interface graphique. Cet outil est à destination de chercheurs ou amateurs de tous horizons (biologistes comme informaticiens). Ce prototype pourrait être grandement amélioré, en collaboration entre l'équipe Mnemosyne (ayant développé l'outil d'annotation [1]), l'équipe à l'IMN étudiant les mécanismes neuronaux de l'apprentissage du chant chez l'oiseau, et des membres de la communauté de recherche internationale ayant également développé un outil d'annotation [2].

## Objectifs du stage

Le stage aura pour objectifs principaux :

1. développer une interface graphique permettant d'entraîner des modèles d'annotation de vocalisations, de visualiser leurs performances et de ré-annoter certaines parties du jeu de données en conséquence (*semi-supervised learning*) ;
2. développer le backend correspondant : gestion des données (audio et annotations), service et persistance des modèles en local (*MLOps*) ;
3. collaborer avec les membres du projet pour définir les besoins, établir les spécifications ou intégrer des outils pré-existants. Cet objectif implique également de collaborer en anglais avec des chercheurs à l'international, et de mettre à disposition du public un outil *open source*.

Le développement sera incrémental : un premier prototype permettra d'entraîner des modèles et de présenter leur évaluation sur l'interface. Une deuxième prototype offrira des possibilités d'édition avancées du jeu de données (ré-annotation de parties de l'audio en fonction des résultats du modèle), et la version finale pourra intégrer des outils d'analyse avancés (détection d'erreur dans le jeu de données, réduction de dimension des spectrogrammes pour de la visualisation voire du clustering, analyse syntaxique des séquences de chant, ...)

## Méthodes

L'étudiant-e devra développer une interface, préférentiellement web, en javascript/typescript (*react...*) ou directement en Python (*bokeh/panel/holoviz...*).

Le backend devra permettre le service de modèles de Machine Learning définis en Python (type *scikit-learn* / *reservoirpy* dans un premier temps, potentiellement type *tensorflow/pytorch*).

L'outil pourra s'inspirer de ou s'intégrer à l'initiative *VocalPy* [3]. L'étudiant-e sera incité à collaborer avec les collaborateurs du projet. Les données pourront par exemple suivre la convention définie par le package *VocalPy crowsetta*.

Une fois terminé, l'outil sera être rendu public, sur *Github*, accompagné de documentation. L'objectif est d'impacter une communauté large et internationale, comme le fait *ReservoirPy* [4], une bibliothèque déjà développée dans l'équipe Mnemosyne pour une communauté ML.

### Compétences requises

- Maîtrise de la programmation web/frontend
- Maîtrise du langage Python
- Bases solides de gestion de projet logiciel
- Connaissances générales en *machine learning* (procédure d'entraînement, d'évaluation, métriques, visualisations, outils Python...)
- Connaissances générales relatives aux réseaux de neurones artificiels

### Candidatures

Date limite : au plus tôt ; dès qu'une bonne candidature est reçue elle sera retenue.

Merci de contacter Nathan Trouvain pour toutes questions, et d'envoyer votre candidature (CV + lettre de motivation) aux trois personnes du projet : Nathan Trouvain, Xavier Hinaut et Arthur Leblois. (Emails en haut de la description de stage)

### Références bibliographiques

[1] N. Trouvain et X. Hinaut, « Canary Song Decoder: Transduction and Implicit Segmentation with ESNs and LTSMs », in ICANN 2021 - 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, sept. 2021, vol. 12895, p. 71-82. doi: 10/gq43sk.

[2] Y. Cohen, D. A. Nicholson, A. Sanchioni, E. K. Mallaber, V. Skidanova, et T. J. Gardner, « Automated annotation of birdsong with a neural network that segments spectrograms », eLife, vol. 11, p. e63853, janv. 2022, doi: 10/gq43sd.

[3] « VocalPy ». <https://github.com/vocalpy>

[3] « ReservoirPy ». <https://github.com/reservoirpy/reservoirpy>