Projet : “*Voice Separation*”
Analyse des données et visualisations

Cursus : DS Juin 21
Membres: Ephi MALGARAS, Mickaël RIVIER, Stany GALLIER

1. Analyse de la base de données MUSDB18

La base de données choisie pour débuter ce projet “séparation de voix” est la base MUSDB18 [1]. Cette base contient 150 pistes de différents genres musicaux pour une durée approximative de 10 h. Chaque piste est également accompagnée de sa décomposition en percussions (“drums”), basse (“bass”), voix (“vocals”) et le reste (“others”), qui inclut notamment tous les autres instruments. La somme “drums”, “bass” et “others” constitue l'accompagnement (c’est-à-dire toute la musique une fois la voix séparée), voir Fig.1.

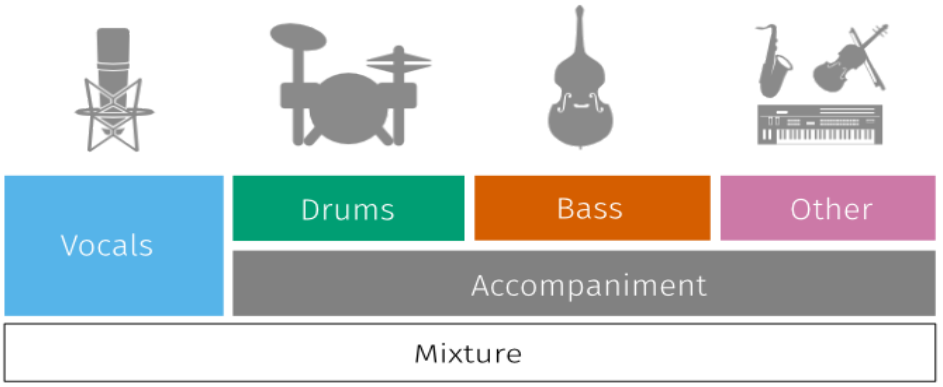

Figure 1. *Une chanson (“mixture”) est décomposée en sa partie voix (“vocals”) et accompagnement (“drums”, “bass” et “others”). Ces quatre parties distinctes (appelées “stems”) ainsi que leur somme (“mixture”) sont disponibles dans la base MUSDB18.*

Les signaux sont stéréophoniques, encodés avec une fréquence de 44.1 kHz et compressés au format mp4, ce qui permet d’avoir une base plus légère (4 Go au lieu de 22 Go pour le format wav).

La base d’apprentissage comporte 100 chansons et la base de test 50 chansons (ratio train/test=67/33). La Fig.2 trace un histogramme des genres musicaux présents dans la base complète. Les différents genres sont représentés de manière peu uniforme avec une surreprésentation des styles pop/rock et rock au détriment de certains styles (jazz, country et reggae notamment).

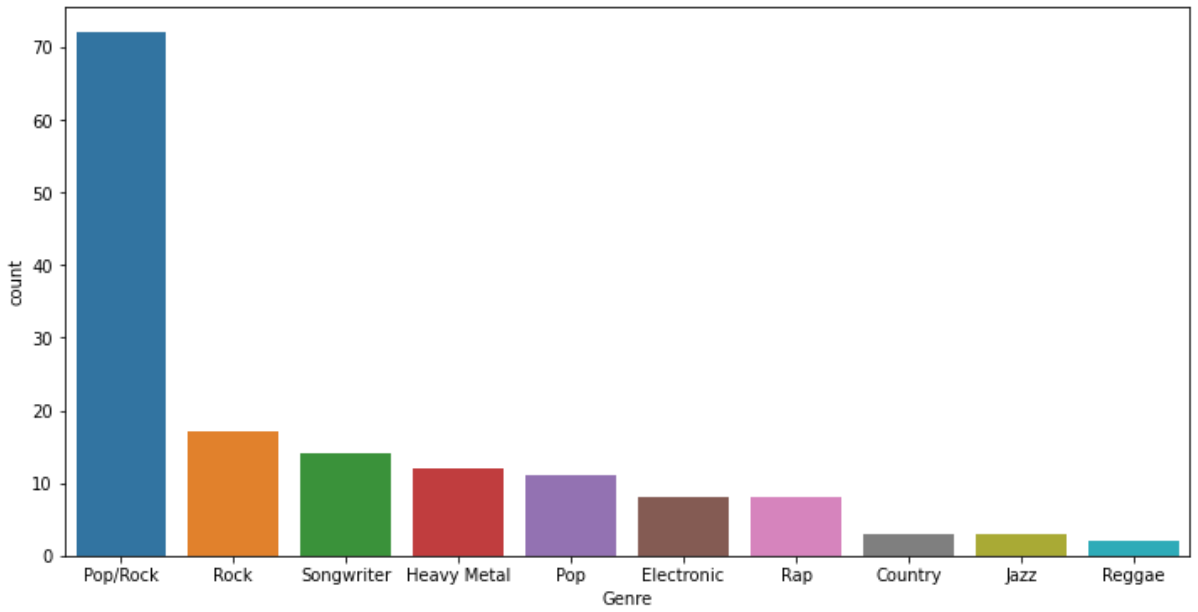

Figure 2. *Répartition par genres musicaux de la base MUSDB18*

On peut vérifier que cette répartition stylistique est similaire dans les sets d'entraînement et de test, pour maximiser la généralisabilité du modèle construit sur les données d'entraînement. Cette visualisation est donnée en Figure 3, avec le set d'entraînement représenté en bleu et celui de test en orange. Il semble que les données de test soient très majoritairement des musiques Pop/Rock, ou le ratio train/test atteint 50/50. On voit dans cette figure que tous les styles sont représentés dans la base d'entraînement, ce qui est un point positif. Cependant, cette omniprésence du style Pop/Rock dans la base de test fait que les métriques finales donneront principalement la capacité de l'algorithme à séparer la voix de l'accompagnement dans un contexte Pop/Rock.

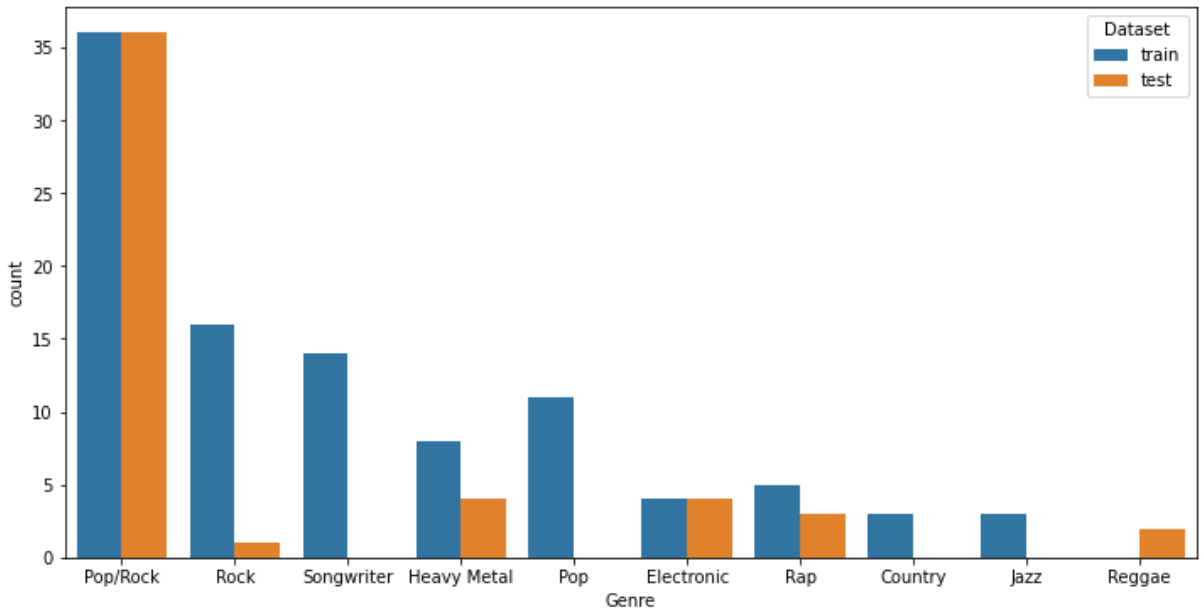

Figure 3. *Comparaison de la répartition des genres musicaux selon la base*

Nous pourrons donc par la suite tester les différents algorithmes non seulement sur le découpage train/test MUSDB18 de référence, mais aussi sur d'autres découpages, mieux répartis stylistiquement. Cela permettra de faire quelques analyses de validations croisées.

Il est aussi possible de visualiser la durée de ces différentes musiques. Un histogramme représentant la distribution de cette durée dans notre base complète, ainsi qu'une estimation KDE, est donné en Figure 4. On peut remarquer que les musiques ont une durée très variables, allant de seulement 12.8 secondes à 10 minutes et 28.3 secondes. La durée moyenne est de 3 minutes et 55.5 secondes, assez proche de la médiane (4 minutes) du fait de la forme relativement simple de la distribution. On remarque finalement le nombre très faible de musiques durant entre 1 et 2 minutes. On en déduit que les musiques présentent sont soit très courtes (moins d'une minute), soit d'un durée d'au moins 2 minutes.

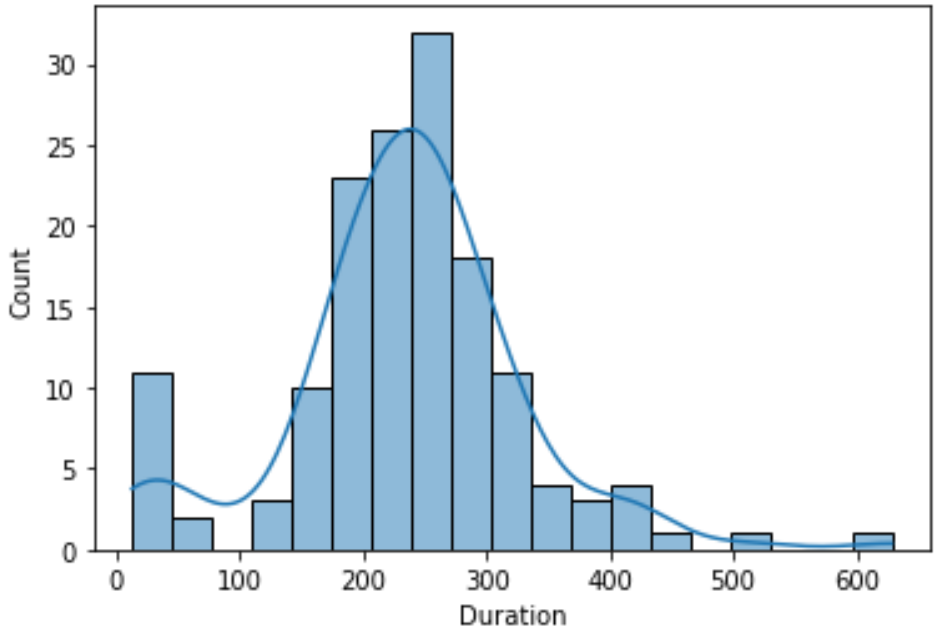

Figure 4. *Histogramme de la durée des musiques, en secondes*

A nouveau, on peut comparer les différences entre la base d'entraînement et la base de test. La Figure 5 superpose les histogrammes de la durée des musiques associées à ces deux bases. On peut remarquer que ces distributions semblent assez similaires, avec une différence de 20 secondes entre les moyennes et seulement 10 secondes entre les médianes. Cependant, il est important de noter que les cas les plus extrêmes (musiques très courtes ou très longues) sont tous inclus dans la base d'entraînement, évitant ainsi le risque de rater une musique "outlier", dont la durée étonnante pourrait aussi être signe d'un style peu banal.

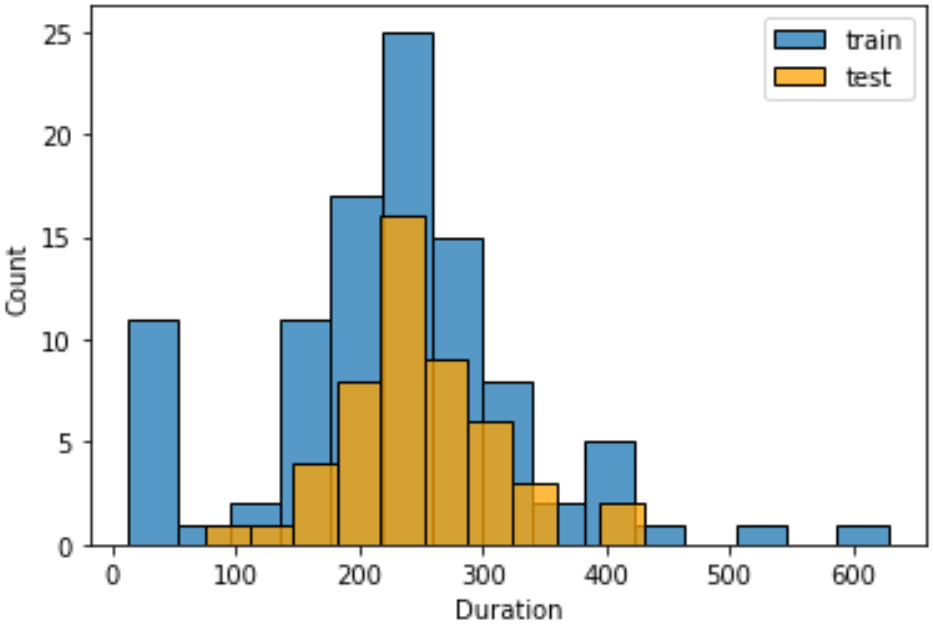

Figure 5. *Histogramme de la durée des musiques en fonction de la base*

Cette base est notamment utilisée lors des évaluations d'algorithmes de séparation de voix (comme le *Signal Separation Evaluation Campaign,* SiSEC). Les données semblent propres, sans valeurs manquantes. Le seul prétraitement éventuel (non encore réalisé ici) est un ré-échantillonnage à une fréquence plus faible (souvent 22 kHz) pour limiter le coût de calcul. Cela permet tout de même d'encoder des sons jusqu'à 11 kHz, suffisant pour la perception humaine.

2. Visualisations des données

Les visualisations (dans l'espace temporel et fréquentiel) sont ici réalisées avec le package `nussl` [2] qui se base notamment sur la bibliothèque `librosa` [3]. Nous traçons ici de courts extraits (7 s.) de chansons issues de la base MUSDB18. En effet, la librairie `musdb` [4], aussi utilisée par `nussl`, permet un rapide téléchargement de ces extraits pour 144 musiques sur les 150 (toutes les musiques de plus de 30 secondes). Cela permet une rapide exploration et visualisation des données. Nous n'avons ici exploré que quelques instances parmi ces 144 chansons.

La Figure 6 trace l'évolution temporelle d'un signal audio (haut), décomposé en ses 4 stems (voix/basse/percussions/autres) au centre puis en voix/accompagnement (bas). L'aspect temporel permet de bien appréhender l'aspect rythmique (assez évident sur la partie basse/percussions). Sur ce cas précis, voix et accompagnement (figure du bas) ont des signatures temporelles différentes mais d'amplitude similaire. L'idée de la séparation de voix est d'extraire la partie vocale (ici en orange) du signal. Puisque la base dispose également d'une décomposition en quatre stems différents, il est envisageable d'extraire séparément ces quatre parties.

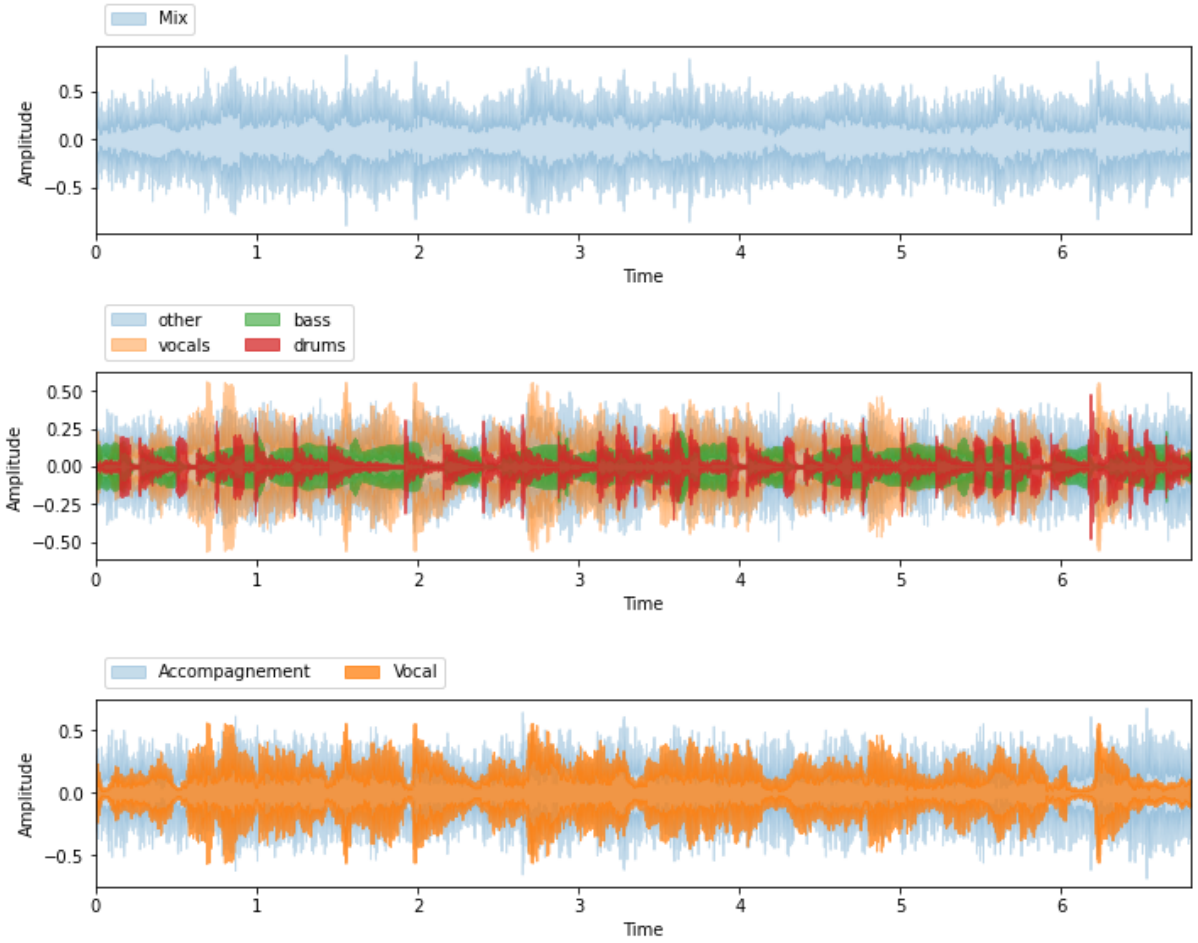

Figure 6. *Signaux temporels : complet (haut), séparés en stem (milieu) ou avec voix extraite (bas)*

Un passage dans l'espace temps-fréquence est plus pertinent pour étudier les caractéristiques spectrales des signaux audio. La librairie `librosa` est ici employée pour tracer des spectrogrammes (temps-fréquence-amplitude). L'amplitude du signal est ici exprimée en dB. La Figure 7 présente deux spectrogrammes de l'extrait précédent, l'un exprimé en fréquence naturelle (Hz), l'autre en Mel [5]. L'échelle de Mel est une transformation logarithmique de la fréquence d'un signal qui rend mieux compte de la perception humaine (par convention 1000 Mel est égal à 1000 Hz). Elle traduit en particulier la difficulté de l'oreille humaine à différencier les hauteurs des sons pour les fréquences élevées.

Les spectrogrammes montrent classiquement des organisations spécifiques (notamment liées aux harmoniques de la voix, comme nous le verrons après). La fréquence maximale est ici vers 20 kHZ comme attendu (échantillonnage à 44 kHz). L'amplitude des signaux est très faible au-delà de ~10 kHz, ce qui peut justifier un ré-échantillonnage en-deçà de 44 kHz (par ex, 22 kHz comme souvent utilisé). L'échelle Mel "comprime" le spectre aux fréquences élevées (10 kHz correspondent à ~3000 Mel) d'où l'absence de signal au-delà de 4000 Mel.

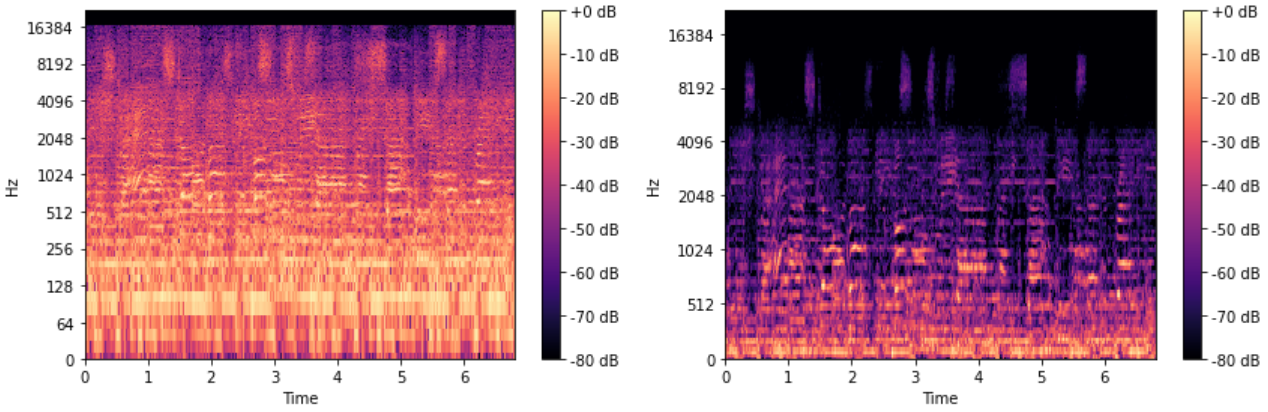

Figure 7. *Spectrogrammes en fréquence (gauche) et Mel (droite). Echelle logarithmique.*

Puisque c'est la voix qui nous intéresse en premier lieu, il est judicieux de tracer un spectrogramme de la voix seule par rapport à l'accompagnement (le "fond musical"). La Figure 8 trace à la fois le mélange (voix+musique), la voix seule et la musique seule ("accompagnement"). Cette fois une échelle linéaire est utilisée pour la fréquence. Le spectrogramme de la voix seule montre une signature spectrale spécifique, avec la présence d'une fréquence fondamentale (vers 200 Hz), accompagnée d'harmoniques (au moins 5 ici) assez nettes. Cette structure diffère assez de l'accompagnement, qui est très riche en fréquences et sans harmoniques visibles (percussions) ainsi que des fréquences très basses (liées à la basse).

Le spectrogramme met ainsi en évidence des spécificités de la voix par rapport à l'accompagnement. Ces spécificités pourront être exploitées dans les traitements futurs. Une des approches classiques est effectivement de travailler dans l'espace fréquentiel à l'aide de masques permettant d'isoler la voix et ses harmoniques.

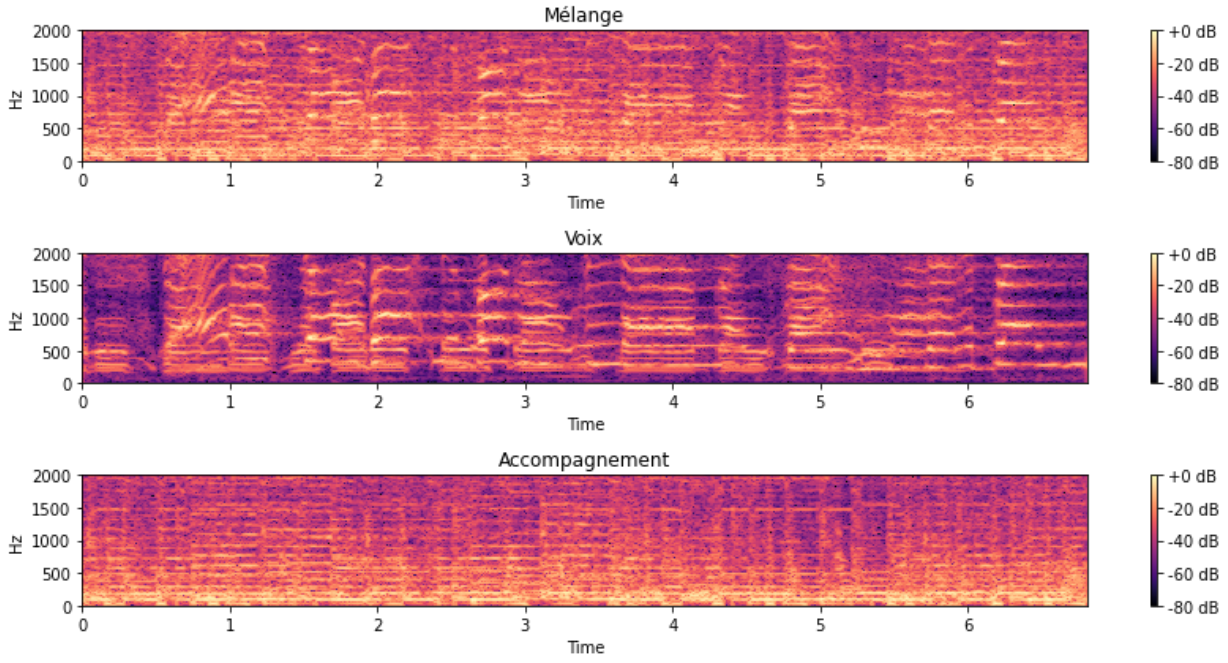

Figure 8. *Spectrogrammes en fréquence : mélange, voix et accompagnement*

Un autre exemple avec des harmoniques encore plus nettes pour la voix est donné en Figure 9. La grosse caisse est aussi particulièrement visible sur le spectrogramme de l'accompagnement, dans les basses fréquences.

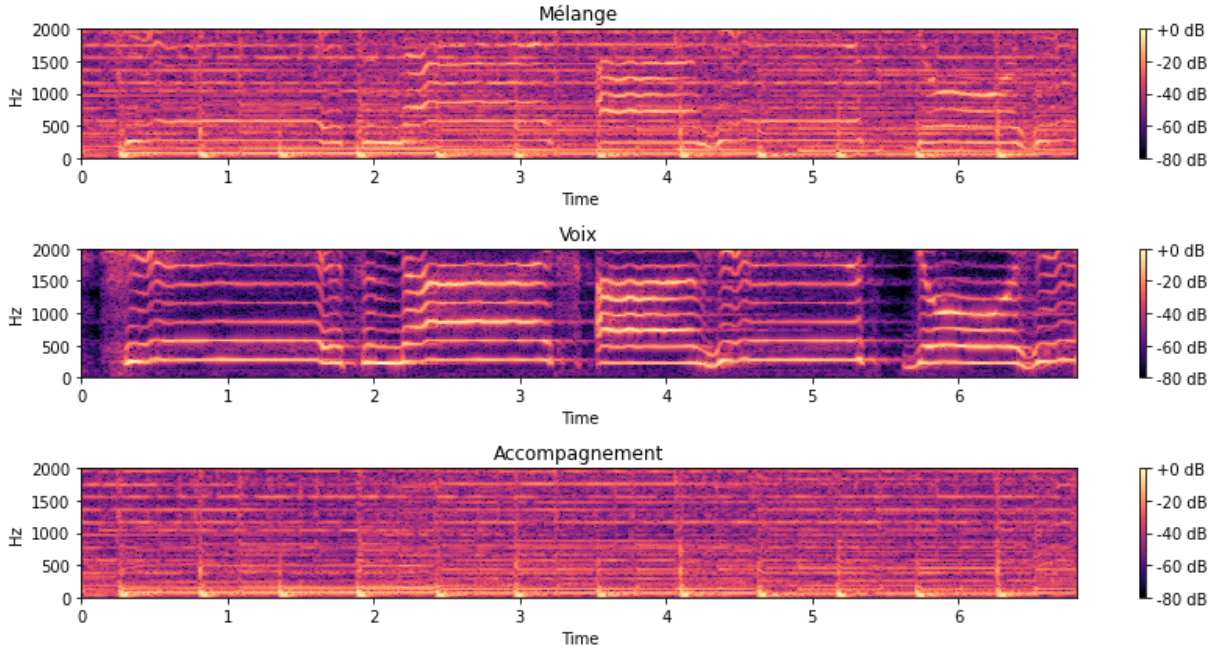

Figure 9. *Autre exemple de spectrogrammes*

On peut finalement noter que chacune de ces pistes, mélangées ou séparées, sont écoutables grâce à la classe `Audio` dans `IPython.display` [6]. Une barre de lecture comme représentée en Figure 10 est alors disponible pour écouter l'extrait.

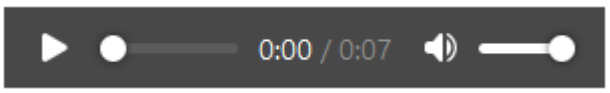

Figure 10. *Barre d'écoute de l'objet IPython.display.Audio*

3. Détection de voix

Une étape préliminaire du projet, préalable à la séparation de voix à proprement parler, peut consister à aborder le problème de détection de voix qui cherche à spécifier si oui ou non un élément sonore est une voix (ou, au contraire, un accompagnement). Il s'agit d'un problème de classification pour lequel il est nécessaire d'avoir une vérité terrain ("*ground truth*"), c'est-à-dire des données labellisées (voix vs. non-voix). Il est ici facilement possible de déterminer ce "masque vocal" puisque nous disposons des données de voix seule.

L'approche proposée ici est assez rudimentaire et consiste simplement à évaluer l'amplitude du signal vocal. Si cette amplitude est au-dessus d'un certain seuil, fixé, alors nous décrétons qu'il y a une voix.

Nous partons du spectrogramme S(f,t) (obtenu par `librosa`) sur la partie vocale uniquement puis la transformons en signal mono (en moyennant les deux pistes stéréo). Ce

spectrogramme est ensuite transcrit en puissance acoustique P et en échelle logarithmique (en dB) via $P(f,t) = 10.log(|S^2|/|S^2|_{max})$. Cette puissance est normalisée par le maximum $|S^2|_{max}$ sur tout le spectre et vaut donc 0 dB au maximum. Pour chaque temps t, nous repérons dans toute la gamme fréquentielle, le maximum de puissance $P_{max}(t) = max_{(0<f<fmax)} P(f,t)$ qui correspond donc au "volume" maximal atteint par la voix au temps t. Cette valeur est ensuite comparée à un seuil fixé (en dB) pour obtenir un masque voix/non-voix. Le seuil ne doit pas être fixé trop bas car même sans voix, il existe toujours une puissance spectrale résiduelle liée à des artefacts issus de la voix (souffle, bruit,...) ou du traitement du signal.

La figure suivante Fig.10 présente un exemple de cette approche avec un spectrogramme de la voix accompagné de son masque vocal temporel. Le seuil est ici choisi à -20 dB. Cette approche mériterait sûrement d'être approfondie notamment par un filtrage préalable ainsi que des critères pour fixer le seuil de manière plus générale.

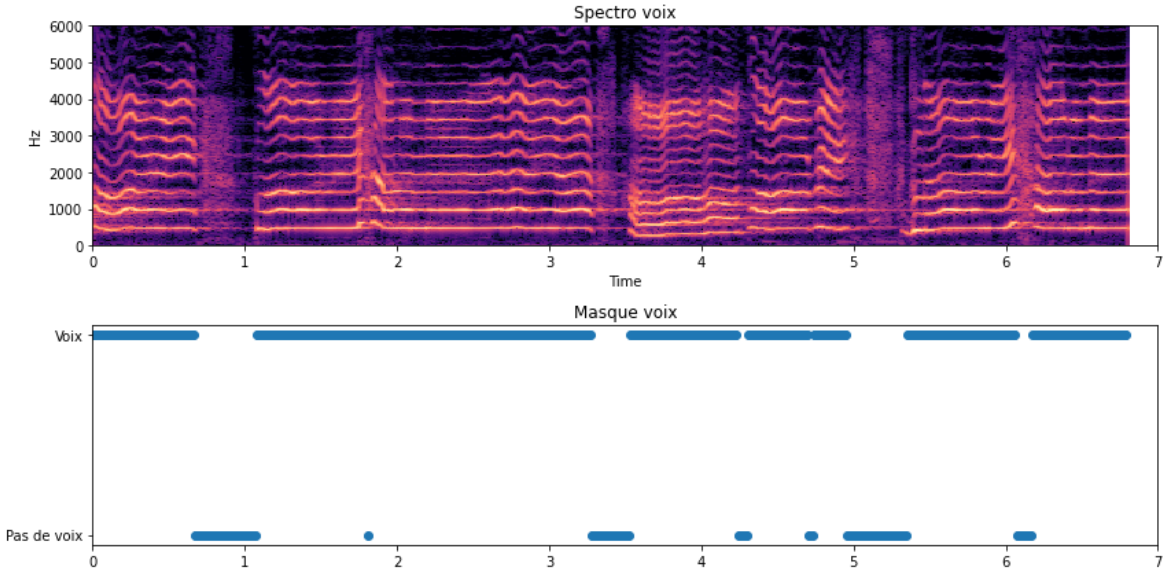

Figure 11. *Exemple de spectrogramme et un masque de présence de voix*

4. Perspectives à court terme

A très court terme, et en attendant la formation sur les réseaux de neurones, indispensable pour mettre au point des approches plus efficaces, nous proposons deux axes de travail :

1. <u>Détection de voix par classification</u>

   L'approche présentée en partie 3 permet de labelliser les données en voix/non-voix. Il devient maintenant possible de mettre en place une approche classique de classification supervisée. Nous proposons d'utiliser des "tranches" temporelles de spectrogramme complet comme *features* afin d'évaluer des approches usuelles (de type forêts aléatoires, régression logistique,...). L'idée est d'évaluer si ces approches simples peuvent "reconnaître" la signature spectrale spécifique de la voix.

2. Utilisation de réseaux neuronaux existants

   En parallèle, nous proposons d'utiliser quelques outils open-source existants, comme Nussl et Open-Unmix, pour conduire une première séparation de voix avec des méthodes à l'état de l'art, bien comprendre les différentes étapes clés et obtenir des résultats de référence, auxquels nous pourrons nous comparer plus tard.

   L'idéal serait d'obtenir rapidement des résultats sur les frameworks les plus connus et les plus performants tels que :

   - Spleeter [7] (conçu par Deezer)
   - Demucs [8] (conçu par Facebook)
   - Open-Unmix [9] (académique, notamment Inria)
   - Nussl [2] (académique, notamment la Northwestern University)
   - Conv-TasNet [10] (académique, notamment la Columbia University)

   Différents benchmarks existent entre ces méthodes, directement sur la base MUSDB18 [11], ou sur le GitHub de Demucs [8].

Bibliographie

1. MUSDB18 database. https://sigsep.github.io/datasets/musdb.html
2. Nussl library. https://github.com/nussl/nussl
3. Librosa library. https://librosa.org/
4. Musdb library. https://github.com/sigsep/sigsep-mus-db
5. Getting to know the Mel spectrogram, *Towards Data Science* https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0
6. IPython Audio display. https://ipython.readthedocs.io/en/stable/api/generated/IPython.display.html
7. Spleeter library. https://github.com/deezer/spleeter
8. Demucs library. https://github.com/facebookresearch/demucs
9. Open-Unmix library. https://github.com/sigsep/open-unmix-pytorch
10. Conv-TasNet library. https://github.com/naplab/Conv-TasNet
11. MUSDB18 benchmark. https://paperswithcode.com/sota/music-source-separation-on-musdb18?p=open-unmix-a-reference-implementation-for