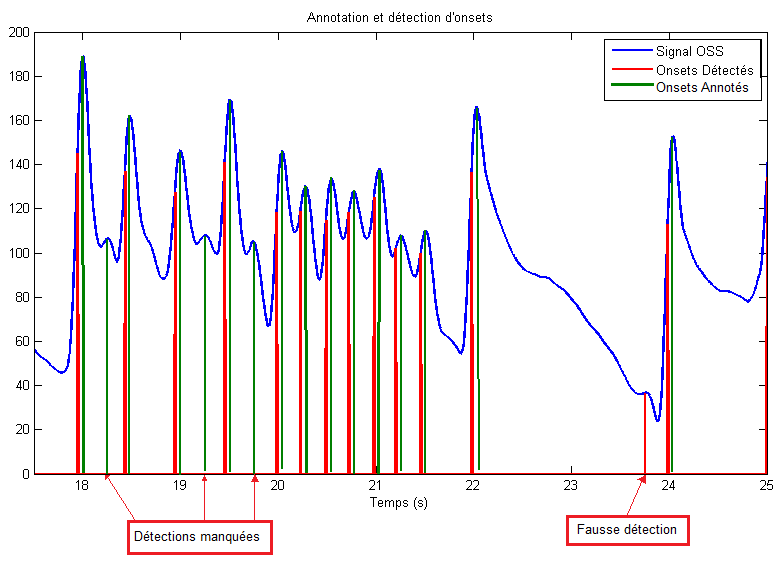
# Méthodes d’évaluation

## Critères d’évaluation

Pour mesurer les performances des différents algorithmes que nous avons développés, nous avons mis en place différents indicateurs spécifique à chaque partie du projet. Ces indicateurs nous ont permis de comparer deux algorithmes ou versions différentes et ainsi sélectionner les meilleures solutions.

Onset Detection :

Concernant l’Onset Detection, nous avons comparé le nombre et les instants des onset détectés comparés avec les onsets attendus. La stratégie est d’appairer les onsets détectés et annotés qui sont proches deux à deux. S’il reste des onsets non appairés, il s’agit de fausses détections ou de détections manquées selon que l’onset seul est respectivement détecté ou annotés. Les offsets ne sont ni annotés ni évalués.



Au final on récupère 3 indicateurs, le taux de fausses détection, le taux de détections manquées et le taux de réussite :

|  |  |
| --- | --- |
| Taux de fausses détections : |  |
| Taux de détections manquées : |  |
| Taux de réussite: |  |
| Où # dénote le « nombre de … » |  |

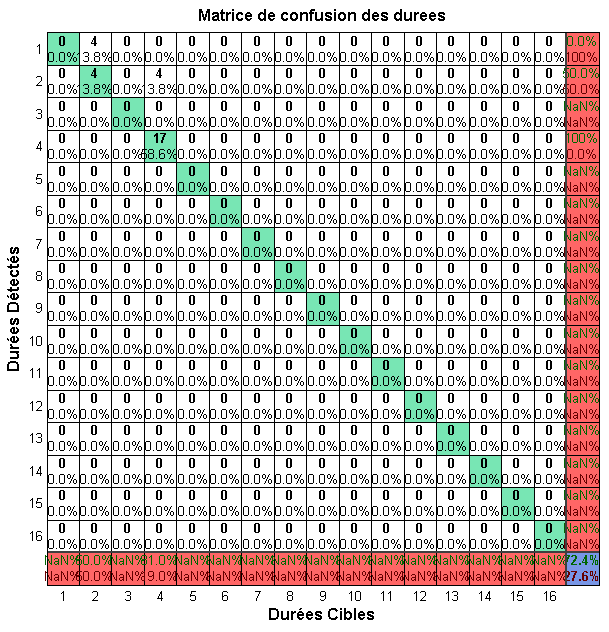
Analyse Rythmique :

Concernant l’Analyse Rythmique, on se base sur le même algorithme que pour l’onset détection au niveau de l’appairage. Pour chaque couple annotation-détection, on compare les durées de la note correspondante. Si cette note est suivie d’un silence car un offset a été détecté directement après, on ajoute la durée du silence à la durée de la note détecté :



Ainsi il n’y a aucun silence à annoter. Les notes issues d’une fausse détection d’onset ne sont pas évaluées, tout comme les détections d’onsets manquées ne sont pas prise en compte.

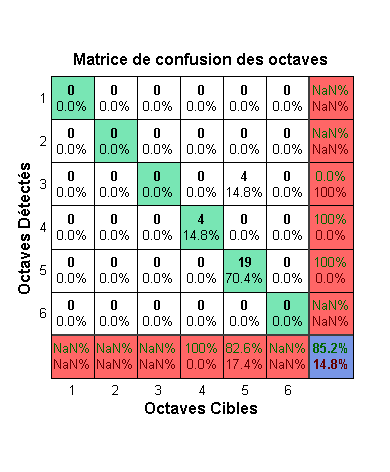
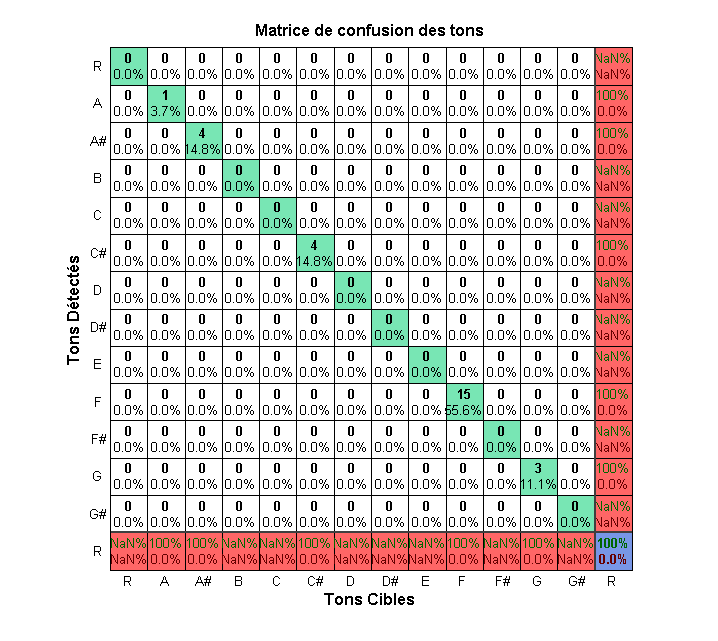
On calcule ensuite la matrice de confusion et utilise le taux de comparaison exacte comme indicateur de performance de l’AR. Cet indicateur est donc relatif au taux de réussite pour l’OD.



Analyse Harmonique :

Concernant l’analyse harmonique, on utilise exactement le même procédé que pour l’analyse rythmique cependant, il n’y a pas de fusion avec les notes « silencieuses », elles sont simplement ignorées. On compare séparément les octaves et les tons (do, do#, ré, etc…).

Là encore on garde pour indicateur, le taux de comparaison exacte de la matrice de confusion. Cet indicateur est lui aussi relatif au taux de succès d’onset detection.



## Constitution d’une base de données annotée

Pour évaluer les performances de nos algorithmes selon les indicateurs décrits précédemment, nous avons besoin de constituer une base de données annotés et répondant à nos critères.

Ajout d’un morceau à la base :

1. Enregistrement d’un son (morceau joué à la guitare) dans des conditions optimales (micro interne du PC relié directement à l’amplificateur). L’enregistrement doit être effectué en rythme (au métronome), sans erreurs ou approximation. Ces conditions sont aussi valable dans le cas où un tiers utiliserait l’application.
2. Rédaction manuelle d’une partition (au format GP4 ou 5) servant de référence. Cette partition correspond au morceau effectivement joué et non au morceau original qui a été reproduit quand c’est le cas.
3. Ajout du morceau aux scripts et lancement de l’algorithme actuel d’onset detection.
4. La sortie de cet algorithme est une liste d’onset erronées. Il faut ensuite manuellement ajouter les onsets manquants et retirer les onsets faussement détectés en se basant sur la partition.
5. Rédaction du fichier « expected.txt » qui sera lu par l’algorithme d’évaluation. Ce fichier est construit comme suit (un élément décrit [] correspond à 1 caractère ou un entier. Les références de solfège sont en notation anglo-saxonne).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ligne | Fichier « expected.txt » | | | | | | |
| 1 | [#notes] | | | | | | |
| 2 | [tempo] | | | | | | |
| 3 | [Échantillon de l’onset n°1] | \t | [durée de la note en nombre de double croche] | \t | [Ton (A…G)] | [# ou espace] | [octave de la note] |
| 4 | Idem pour la note n°2 | | | | | | |
| Etc… | | | | | | | |

Toutes ces manipulations prennent du temps à effectuer (environ une demi-journée par morceau).

C’est la raison pour laquelle notre base de données est si restreinte. D’autre part, avant la mise en place de cette méthode et notamment l’étape n°1, nous avions inséré dans notre base un morceau issu directement de la version CD (les 8 premières secondes de Day Tripper où uniquement de la guitare est jouée.) Cependant ce morceau est trop court pour qu’une bonne estimation du tempo soit faite. Nous avions également inséré des morceaux générés à partir de la banque de sons du logiciel Guitar Pro. Ces sons sont synthétiques et les partitions sont très simples. Notre logiciel obtient donc des résultats presque parfait (aux différences d’interprétation près comme le tempo). Cela prouve qu’en théorie les solutions choisies sont les bonnes et que les erreurs sont introduit par l’instrument réel et le musicien réel (différences d’attaques, de timbres, d’enveloppe, erreurs de rythme, etc…). Notre logiciel est donc capable d’effectuer l’opération inverse de Guitar Pro à savoir : « créer une partition à partir d’un son synthétique ». Ces bons résultats ont été obtenus tôt dans notre projet, et notre travail par la suite a été d’améliorer les performances sur des morceaux réels.