

# MANUEL D'UTILISATION DU 'CLASSEUR-CODEUR PFC (SCHWA ET LIAISON) EN PERL'

Version 1.0 (janvier 2004)

Jacques Durand, Julien Eychenne et Abderrahim Meqqori

## 0. Introduction

Le programme présenté ici vise à extraire, classer et interroger les codages effectués dans le cadre du projet PFC pour le schwa et la liaison. Ce programme est différent des outils déjà développés par A. Meqqori (classeur schwa et classeur liaison). En effet, ces derniers sont des outils autonomes et dédiés à ces tâches. Ils sont donc très conviviaux mais ne sont pas extensibles. Par ailleurs, ils fonctionnent seulement sur deux plateformes : Windows et MacOS.

Le programme décrit ici et qui sera appelé 'Classeur-codeur' fonctionne normalement sur toutes les plateformes qui disposent de l'interpréteur 'Perl' (et en particulier, Linux, Unix, Windows, MacOS). Perl est un langage de programmation disponible sous une licence libre. Bien que le 'Classeur-codeur' exige une installation préalable de Perl, il ne présuppose pas la connaissance de ce langage de programmation. En revanche, il faut noter que le 'Classeur-Codeur' n'interagit pas avec l'utilisateur au moyen d'une interface graphique classique comme celle de Windows mais nécessite le recours à des lignes de commande de type DOS sous Windows ou 'shell Bash' sous Linux et MacOS X. Ce manuel commencera donc par l'installation de PERL sous Windows en 1 avant d'explorer l'utilisation du Classeur-codeur en 2.

## 1. Installation

### 1.1. Installation sous Windows

#### 1.1.1. Installation de PERL

L'outil PERL est gratuitement téléchargeable à partir de [www.activestate.com](http://www.activestate.com). Pour être plus précis, le téléchargement s'effectue à l'adresse suivante en fonction de votre plateforme : <http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl>.

Sauf information contraire de notre part, on préférera la version la plus récente. Les utilisateurs de Windows veilleront à télécharger AS Package (et non MSI qui nous a posé des problèmes<sup>1</sup>).

Une fois téléchargé, il faut décompresser le fichier *ActivePerl-x.x.x.806-MSWin32-x86.zip* (par exemple *ActivePerl-5.8.0.806-MSWin32-x86.zip* au moment de la rédaction de ce manuel). Attention, sur certains systèmes, les extensions de type .zip, .exe, etc. ne sont pas visibles). La décompression créera un dossier nommé *ActivePerl-x.x.x.806-MSWin32-x86* contenant lui-même un dossier *ActivePerl* et un exécutable nommé *Installer.bat*. Il faut prévoir au moins 35 Mo d'espace libre.

---

<sup>1</sup> Il faut noter que le paquet MSI de la version 5.8.2 de Perl disponible sur [www.activestate.com](http://www.activestate.com) semble désormais fonctionner correctement sur Windows 2000/XP. L'utilisateur pourra donc le choisir sans crainte s'il souhaite disposer d'un installateur plus « classique » (avec une interface graphique).

A ce stade, il faut double-cliquer sur *Installer.bat*, ce qui ouvre une fenêtre d'installation dont les éléments sont listés ci-dessous (sauf mention contraire, il suffit d'appuyer systématiquement sur la touche entrée pour valider les choix par défaut) :

```
Enter top level directory for install [c:\Perl]:
ActivePerl 806 will be intalled into 'c:\Perl'
Proceed? [y]
If you have a development environment (e.g. Visual Studio)...
Proceed? [y]
Creat shortucuts to the HTML documentation? [y]
Add the Perl/bin directory to the PATH? [y]
```

Si l'installation s'est déroulée correctement, Perl devrait être l'un des répertoires sous C: et vous pouvez maintenant installer le classeur-codeur PFC.

### 1.1.2. Installation du logiciel proprement dit

1.1.2.1. Copiez sur votre ordinateur le dossier classeur-codeur.zip (qui est sur le site PFC ou qui vous a été transmis)<sup>2</sup> et décompressez-le, ce qui fournira un dossier classeur-codeur. La localisation de ce dernier sur votre ordinateur est sans importance. (Vous pouvez effacer le dossier zip une fois cette opération faite.)

1.1.2.2. Rentrez dans le dossier classeur-codeur et double-cliquez sur *Install-classeur.bat*, ce qui installera les éléments nécessaires pour le fonctionnement de l'outil.

1.1.2.3. Le fichier *classeur(.bat)* est désormais utilisable comme tout programme exécutable. Vous pouvez le laisser dans le dossier *classeur-codeur* ou le déplacer.

## 1.2. Installation sous MacOS (version antérieure à MacOS X)

### 1.2.1. Installation de MacPerl

L'outil MacPerl est gratuitement téléchargeable à partir de [www.macperl.org](http://www.macperl.org). Plus exactement, le programme lui-même peut être téléchargé à partir de [www.sourceforge.net](http://www.sourceforge.net) (mettre l'URL). Il faut y récupérer la dernière version de MacPerl (5.6.1 à l'heure où nous écrivons cette documentation), en choisissant un miroir de téléchargement proche. Attention, le fichier à télécharger pèse environ 20 Mo ! Le fichier se nomme *macperl\_5\_6\_1r2\_full.bin*. Une fois le fichier téléchargé, il convient de l'ouvrir avec un outil tel que Stuffit Expander ou MacBinary, qui se chargera d'extraire l'installateur (*MacPerl Full Installer*). Il faut ensuite double-cliquer sur l'installateur pour lancer le processus d'installation. Il faut se laisser guider par les fenêtres successives jusqu'au bouton « Install » qui installera MacPerl sur le disque contenant le système (il est possible de choisir un autre disque). Une fois l'installation terminée, il suffit de cliquer sur le bouton « quit » : un dossier « MacPerl f » a été créé à la racine du disque précédemment sélectionné.

### 1.2.2. Installation du classeur-codeur proprement dit

Copiez sur votre ordinateur le dossier *classeur-codeur.zip* (qui est sur le site PFC ou qui vous a été transmis) et décompressez-le à l'aide de Stuffit Expander ou d'un autre utilitaire ZIP, ce qui fournira un dossier classeur-codeur. La localisation de ce dernier sur votre ordinateur est sans importance. (Vous pouvez effacer le dossier zip une fois cette opération

---

<sup>2</sup> Ou toute autre archive qui vous aura été fournie (p.ex. 'outils\_perl\_2004.zip').

faite.) Rentrez dans le dossier classeur-codeur, où se trouve l'application *classeur.mac* : elle est utilisable comme toute application Mac (il suffit de double-cliquer dessus pour l'ouvrir).

### **1.3. Installation sous UNIX (GNU/Linux, MacOS X, etc.)**

Contrairement aux autres plateformes, il est extrêmement peu probable que vous ayez à installer Perl, puisqu'il est généralement fourni en standard sur les systèmes UNIX. Pour vérifier que Perl est déjà présent sur votre système, tapez la commande 'which perl' ou 'locate perl' qui doit vous renvoyer une réponse du type « /usr/bin/perl » (variable selon les installations). Si la commande vous rend la main sans donner de réponse, cela signifie que Perl n'est pas installé sur votre système. S'il s'agit d'un système dont vous n'êtes pas l'administrateur (cas typique d'une machine en réseau dans une université), vous devez contacter l'administrateur système et lui demander d'installer Perl. S'il s'agit d'un système UNIX dont vous êtes l'administrateur (p.ex. une installation sur votre propre ordinateur), vous devrez l'installer vous-même avec les outils spécifiques à votre système UNIX (perl est généralement présent sur les CD d'installation). Il n'est pas possible, dans le cadre de cette documentation, de fournir une explication détaillée de l'installation, étant donnée la variété des implémentations d'UNIX. En cas de problème, adressez un courriel à Julien Eychenne en précisant votre système d'exploitation.

Nous supposons que l'outil vous a été remis sur disquette et présentons une méthode d'installation sous Linux (les distributions récentes permettent de copier les fichiers avec la souris). Dans le terminal, tapez la série de commandes suivantes :

```
cd
mkdir disktmp outilsPFC
mount -t auto /dev/fd0 disktmp
ls disktmp
tar xzf disktmp/classeur-codeur-x.x.tar.gz
umount disktmp
rmdir disktmp
```

Vous devrez remplacer x.x dans classeur-codeur-x.x.tar.gz par le numéro de version, indiqué par la commande précédente. A ce stade, vous disposez d'un dossier classeur-codeur dans votre répertoire personnel (*home directory*). Nous allons copier le programme à la racine du répertoire personnel (symbolisé par « ~ » ou « \$HOME »), puisque c'est le répertoire dans lequel nous nous trouvons :

```
cp classeur-codeur/src/classeur.pl ~/outilsPFC
```

A partir de là, le classeur-codeur peut être lancé par la commande : « perl outilsPFC/classeur.pl », ou « perl /chemin/vers/classeur.pl » si vous n'êtes pas dans le dossier \$HOME.

## **2. Utilisation du classeur-codeur**

Au lancement de l'application, une fenêtre proposant les choix suivants apparaît :

*Que souhaitez-vous faire ?*

- *Formater et traiter une base PFC anonymisée (touche f)*
- *Extraire toutes les occurrences du schwa (touche e)*
- *Extraire les codes liaison (touche l)*

- *Effectuer une requête en utilisant la syntaxe des expressions rationnelles Perl<sup>3</sup> (touche r)*
- *Contacteur l'auteur (touche i)*
- *Quitter le programme (touche q)*

Avant que d'aborder chacune de ces fonctions (points 2.2. et suivants), nous présenterons en 2.1. les particularités propres à chaque système d'exploitation pour la saisie des chemins d'accès.

## **2.1. Les chemins d'accès sur les différents systèmes d'exploitation**

### **2.1.1. Windows**

Pour les utilisateurs qui ne sont pas habitués aux lignes de commande DOS, nous rappellerons que l'accès aux répertoires et/ou aux fichiers est normalement exprimé par rapport à la racine du disque qui les contient. S'il n'y a qu'un seul disque dur, ce sera C: . Donnons un exemple concret : la base à traiter est le répertoire *vendee* situé immédiatement sous la racine. Dans ce cas, la ligne de commande représentant ce dossier serait *C:\vendee*, où l'antislash (\) indique une relation de contenant à contenu. Prenons maintenant un exemple plus complexe : la base à traiter est le dossier *vendee*, lui-même situé dans un dossier PFC à la racine. La commande sera alors *C:\PFC\vendee*. Cette façon maximale de spécifier le nom d'un fichier ou dossier est appelée chemin absolu (par opposition au chemin relatif, qui n'est pas utilisé par notre programme). Pour réduire la complexité des commandes à saisir, nous suggérons de créer un dossier PFC à la racine du disque, dans lequel seront stockés les points d'enquête. Cependant, si vous ne traitez qu'un seul point d'enquête, il est plus judicieux de le placer directement sous la racine.

Enfin, dans la mesure où Perl est sensible à la casse, on respectera scrupuleusement le format typographique des fichiers et dossiers (majuscules et minuscules). De même, afin de garantir une compatibilité maximale entre les systèmes d'exploitation, on évitera les accents et les espaces dans les noms de fichiers et de dossiers.

## **2.2. Formater et traiter une base PFC anonymisée (touche f)**

### **2.2.1. Choix de la base à traiter**

Taper *f* suivie de la touche entrée, le message suivant apparaît : *Chemin \*absolu\* vers le dossier du point d'enquête :*

Exemple : *C:\PFC\vendee*

Pour valider, taper sur la touche entrée. Le message suivant apparaît : *Veillez patienter quelques instants...*

Après quelques instants, suivant la vitesse de la machine et la taille de la base, la fenêtre se referme, ce qui signale la fin du traitement. Attention, étant donné que le programme crée un dossier *analyse*, il est recommandé d'effacer tout dossier *analyse* précédemment créé par le classeur.

### **2.2.2. Consultation des résultats**

L'outil crée dans le dossier spécifié, *vendee* dans notre exemple, un sous-dossier nommé *analyse*, dans lequel il aura créé un dossier pour chaque locuteur se trouvant dans la base de

---

<sup>3</sup> Voir explication dans l'alinéa 2.4.

départ et un dossier nommé 000 qui représente l'analyse de la base entière. L'avantage du classeur est donc de permettre l'analyse de résultats à la fois au niveau individuel et au niveau collectif.

Exemple concret :

Prenons une base contenant les locuteurs suivants :

- 11aal1
- 11agm1
- 11adp1
- 11ajp1

Dans cette base, prenons le locuteur 11aal1. Les fichiers pertinents pour le classeur sont ceux qui ont été codés pour le schwa et la liaison, à savoir :

- 11aal1gg.Textgrid : entretien guidé
- 11aal1lg.Textgrid : entretien libre
- 11aal1tg.Textgrid : texte

Le classeur traite ces trois fichiers pour le schwa et la liaison, mais fournit aussi d'autres résultats qui peuvent intéresser l'analyste. Ainsi, pour chaque locuteur, il crée un fichier conversation (*11aal1cg.textgrid* dans notre exemple) correspondant à la concaténation des entretiens libre et guidé, qu'on pourra comparer à la lecture de texte. Il crée aussi un fichier global (ici *11aal1leg.textgrid*) qui rassemble les codages des conversations libre et guidée ainsi que du texte. Pour chaque locuteur, le programme produit :

- un dossier *donnees* contenant les fichiers TextGrid de départ et les concaténations pertinentes pour nos analyses, soit 5 fichiers au total.
- un dossier *liaison* contenant le classement des codes liaison pour les cinq fichiers : fichier entretien libre, fichier entretien guidé, fichier texte, fichier conversation (guidé + libre), fichier global (guidé + libre + texte).
- un dossier *schwa* contenant le classement des codes schwa ( et quelques éléments de statistiques) pour les cinq fichiers : fichier entretien libre, fichier entretien guidé, fichier texte, fichier conversation (guidé + libre), fichier global (guidé + libre + texte).

Prenons un exemple concret : les 5 fichiers du classement du schwa pour le locuteur *11aal1* seront :

- *11aal1cg.TextGrid\_schwa.txt* (classement du codage schwa pour la conversation guidée + libre)
- *11aal1leg.TextGrid\_schwa.txt* (classement du codage schwa pour la conversation guidée + libre + lecture de texte)
- *11aal1gg.TextGrid\_schwa.txt* (classement du codage schwa pour la conversation guidée)
- *11aal1lg.TextGrid\_schwa.txt* (classement du codage schwa pour la conversation libre)
- *11aal1tg.TextGrid\_schwa.txt* (classement du codage schwa pour la lecture de texte)

Le dossier supplémentaire 000 contient quant à lui les mêmes éléments que le dossier d'un locuteur, à l'échelle de toute la base. On y trouvera un dossier *donnees*, un dossier *liaison* et un dossier *schwa*. Pour être exhaustif, le dossier *schwa* inclura pour toute base les fichiers suivants :

- *000cg.TextGrid\_schwa.txt* (classement du codage schwa pour la conversation guidée + libre)

- 000eg.TextGrid\_schwa.txt (classement du codage schwa pour la conversation guidée + libre + lecture de texte)
- 000gg.TextGrid\_schwa.txt (classement du codage schwa pour la conversation guidée)
- 000lg.TextGrid\_schwa.txt (classement du codage schwa pour la conversation libre)
- 000tg.TextGrid\_schwa.txt (classement du codage schwa pour la lecture de texte)

Le même principe de dénomination est adopté pour la liaison.

### 2.3. Extraire toutes les occurrences du schwa (touche e)

Contrairement à l'option précédente qui traite une base entière de manière systématique, cette option permet de spécifier le fichier à traiter pour extraire toutes les occurrences du schwa. La marche à suivre est la suivante :

- Appuyer sur la touche e suivie de la touche entrée pour lancer le processus
- Le message suivant apparaît : *Chemin absolu vers le fichier TextGrid à traiter.*
- Indiquez le chemin absolu du fichier (voir plus haut) : par exemple C:\douzens\11aal1\11aal1gg.Textgrid.

- Le message suivant apparaît :

*Le fichier de sortie est un fichier texte (lisible avec un éditeur de texte), et doit porter l'extension .txt*

*Quel nom souhaitez-vous lui donner ("sortie\_classeur.txt" par défaut) ?*

Deux cas de figure se présentent :

- appuyer sur la touche entrée : le fichier de sortie portera le nom « sortie\_classeur.txt ».
- taper un nom (sans espaces) portant l'extension .txt (ex: mon\_fichier.txt) puis la touche entrée.

Attention : si vous tapez un nom sans extension (ex : 'mon\_fichier') ou avec une extension autre que .txt (ex : 'mon\_fichier.htm'), le fichier de sortie portera le nom par défaut ('sortie\_classeur.txt'), ceci afin d'éviter aux utilisateurs de Windows certains problèmes liés à la reconnaissance des extensions.

Après un moment d'attente, affiche le message suivant : « *Bonne analyse...* », signifiant ainsi la fin du traitement. Les résultats peuvent alors être consultés dans le fichier de sortie (créé au même niveau que le programme, par exemple le dossier CLASSEUR-CODEUR).

Les résultats se présentent sous la forme suivante :

```
#####
### MONOSYLLABE ###
#####

*****
0111 ---> 2
*****

0 : schwa absent
1 : monosyllabe
1 : voyelle à gauche
1 : voyelle à droite
*****
1 inique ? DP: Alor0412s moi je0111 suis née en soixante1411 et d
```

2 2 gauche1413.' GM: Et ben, je0111 lui ai dit 'n'en parlons plus

\*\*\*\*\*

1111 ----> 4

\*\*\*\*\*

1 : schwa présent

1 : monosyllabe

1 : voyelle à gauche

1 : voyelle à droite

\*\*\*\*\*

1 . Donc0412 ce0122 qui fait que1111 à l'âge0412 de1121 onze0411 a  
2 bien d'enfants? ML: J' ai que1111 Anne1413, <E: Tu as> j'ai que  
3 Anne1413, <E: Tu as> j'ai que1111 Anne1414 <E: que Anne,> qui a  
4 rcaonne.> bon je2112 mets je1111, une1432 bonne1412 route1412

%%% contexte x111 % % %

- schwa est absent dans 33.33% des cas, soit 2 occurrence(s).
- schwa est présent dans 66.66% des cas, soit 4 occurrence(s).
- schwa est incertain dans 0% des cas, soit 0 occurrence(s).

## 2.4 Extraire les codes liaison (touche l)

Cette option fonctionne de la même manière que l'option précédente (cf. 2.2) : la manière de procéder est la même que pour le schwa, à ceci près que le nom par défaut du fichier de sortie est sortie\_liaison.txt (et non sortie\_schwa.txt).

\*\*\*\*\*

11t ----> 14

\*\*\*\*\*

1 : mot liaisonnant monosyllabique

1 : liaison enchaînée

t : consonne liaisonnante (valeur SAMPA)

\*\*\*\*\*

1 té la maison, AL: euh qui est1 1t à côté de la mienne, et puis d  
2 re était d'ici. AL: Et quand1 1t on s'est partagé avec mes soeu  
3 euh AL: à Aiguevives qui est1 1t à neuf kilomètres d'ici. AL:  
4 ça et j'ai arraché, j'ai tout1 1t arraché AL: et je me suis rec  
5 n. AL: J'aurais continué peut1 1t être c'est possible. AL: Mais  
6 eurs c'est, j'ai demandé quand1 1t on s'est partagé un salaire di  
7 un salaire différé. AL: C'est1 1t à dire que moi, de dix sept an  
8 sins debout, AL: mon père est1 1t un peu comme moi, il a travail  
9 is d'août pensant que bon tout1 1t allait se passer20 à ce moment  
10 ce moment-là. AL: Et le vingt1 1t huit, ou vingt neuf août là, f  
11 res recommandations20 et quand1 1t il a eu fini (pff) AL: il a (  
12 aetera AL: puis10 alors quand1 1t il a eu fini clac, cloc. AL:  
13 ris l'ampoule et AL: il s'est1 1t aperçu que c'était20 un truc e  
14 intenant et puis clac il s'est1 1t éteint. AL: Pouf, un souffle

\*\*\*\*\*

11nVN ----> 1

\*\*\*\*\*

1 : mot liaisonnant monosyllabique

1 : liaison enchaînée

n : consonne liaisonnante (valeur SAMPA)

VN : liaison enchaînée en [n] précédée d'une voyelle nasale

\*\*\*\*\*

1 re tout ça puis j'avais20 u/-n1InVN autre beau-frère qui était c

### **2.5. Effectuer une requête en utilisant la syntaxe des expressions rationnelles Perl (touche r)**

Les expressions rationnelles (ou expressions régulières, en anglais *regular expressions*, abrégé en *regexp*) sont des commandes dans une syntaxe permettant de rechercher des suites de caractères dans des fichiers de texte. L'intérêt de cette syntaxe est qu'elle permet de trouver des suites de caractères répondant à certains critères choisis par l'utilisateur. Ainsi, on peut chercher tous les *e* graphiques suivis du chiffre 0, ou de 1 ou de 2, et suivis des chiffres 4 et 1 et 2. Cette requête aura la forme : `e[0-2]412`.

Cette fonction représente une utilisation avancée de l'outil, et nous renvoyons donc à la documentation de Perl (`perlre` et `perlretut`) pour aller plus loin.

Il n'est pas exclu qu'une interface plus simple pour les requêtes fréquentes soit implémentée dans une version ultérieure de l'outil.

### **2.6. Contacter l'auteur (touche i)**

Donne l'adresse mail de l'auteur afin de signaler tout problème ou toute suggestion pour des améliorations futures.

### **2.7. Quitter le programme (touche q)**

Quitte le programme.