INFORMATICA MUSICALE

UNIVERSITA' DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
LAUREA TRIENNALE IN INFORMATICA
A.A. 2018/19
Prof. Filippo L.M. Milotta

ID PROGETTO: 0B

TITOLO PROGETTO: Flac & Mp3: Lossless e Lossy a confronto

AUTORE: Costanzo Giacomo

Indice

1. Obiettivi del progetto

Il progetto si pone l'obbiettivo di confrontare la stessa traccia audio in formati diversi, nel nostro caso flac e mp3 (con più opzioni), per vedere (e sentire) le differenze tra i brani.

Per il confronto della fase utilizzo Audacity, invertendo la fase della traccia codificata con i tool e mixandola con la traccia originale (formato wave). Invertendo la fase, in caso le due tracce audio siano uguali, la traccia risultante dovrebbe essere silenziosa, questo perché due onde sonore in opposizione di fase (a parità di frequenza) si elidono a vicenda.

Tramite MATLAB calcolo i PSNR e MSE rispetto alla traccia originale, solitamente sono due metodi utilizzati per le immagini, ma posso adattarsi pure ai segnali audio senza problemi.

il Mean Square Error (MSE) non è altro che l'errore quadratico medio, indica una differenza quadratica dei valori osservati rispetto agli originali. Un basso valore indica due immagini o tracce audio più simili tra di loro, un valore elevato indica una traccia con valori diversi rispetto all'originale, nel nostro caso più le tracce audio sono simili, più il valore si avvicinerà allo zero.

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[I'(x,y)$$

Il Peak Signal-to-noise ratio (PSNR) è il rapporto tra la potenza massima del segnale e del rumore che può compromettere la fedeltà della traccia compressa, anch'esso utilizzato solitamente per le immagini, il modo più semplice per definirlo è tramite l'MSE. Più il valore sarà alto, più il file compresso sarà fedele rispetto all'originale

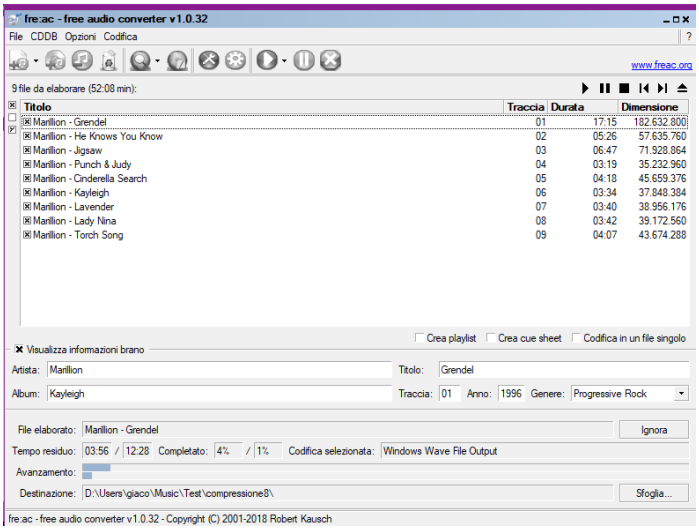
$$PSNR = -10log_{10}\frac{MSE}{S^2}$$

Nel caso di file identici MSE=0, mentre il PSNR= ∞.

2.0 Metodo Proposto (Elenco Breve)

1. Ripping dei file da cd
2. Conversione nei formati flac, mp3 128, abs 320, cbr 320
3. Riconversione dei formati lossy nel formato originale (per il confronto)
4. Confronto di ogni singola traccia con audacity mediante:
    a) Importazione della traccia estratta da cd
    b) Importazione della traccia convertita e sua inversione della fase
    c) Mix in una nuova traccia
5. Calcolo del PSNR e MSE tramite MATLAB

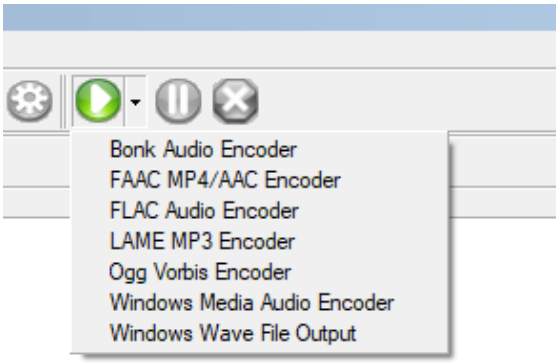
2.1 Metodo Proposto



Per il ripping dei file è stato usato free:ac, il programma una volta inserito il CD, analizzerà le tracce, con la possibilità di scegliere la codifica preferita (in questo caso Windows Wave File Output).

Una volta scelte le tracce, basta, cliccare nella freccia verde, sotto la barra degli strumenti per avviare la codifica (oppure si può aprire una tendina dal menù per poter scegliere quale codifica utilizzare).

Una volta ottenuta la traccia bisogna utilizzare degli strumenti appositi per la conversione del formato, la mia scelta è ricaduta sui tool LAME e Flac Tool.



Il FLAC è l'acronimo di Free Lossless Audio Codec, l'ultima versione è la 1.3.2 ed è nato da un'idea di Josh Coalson del 1999, è un formato audio Lossless utilizzato principalmente per la compressione, il progetto è Open Source.

L'**mp3** è l'abbreviativo di MPEG1-Layer III, definito nel 1992, supporto un massimo di 2 canali con un bitrate di 320kb/s massimo. Lo standard definisce la qualità di codifica mp3 per i cd a 128 kb/s ma come si vedrà dai test questa qualità non è sufficiente per mantenere la stessa qualità di quest'ultimo. Brevettato da Thomson e Fraunhofer, le informazioni rimosse durante la compressione variano in base all'encoder utilizzato, ma si basano sulla psicoacustica (si cercano di mantenere i suoni che si sentono meglio nell'orecchio umano) in modo da poter comunque percepire l'informazione correttamente anche in presenza di dati mancanti.

Il **Flac Tool** è lo strumento utilizzato per convertire i brani in formato Flac, al suo interno è presente sia la possibilità di fare sia la codifica, che la decodifica, si utilizza tramite terminale. Il comando di default è *flac "nome_file"* che darà in output un file flac con livello di compressione 5. La compressione utilizzata può variare da 0 a 8, è indica di quanto il file viene compresso (a discapito della velocità di encoding), per scegliere un livello di compressione posso utilizzare *flac -8 "nome_file"* dove ad 8 posso sostituire il livello desiderato, posso pure convertire un'intera cartella utilizzando il comando *flac *.wav*

```
PS D:\Users\giaco\Desktop\flac-1.3.2-win\win64> D:\Users\giaco\Desktop\flac-1.3.2-win\win64\flac.exe -5
 "D:\Users\giaco\Desktop\flac-1.3.2-win\win64\01 Grendel.wav"
flac 1.3.2
Copyright (C) 2000-2009  Josh Coalson, 2011-2016  Xiph.Org Foundation
flac comes with ABSOLUTELY NO WARRANTY.  This is free software, and you are
welcome to redistribute it under certain conditions.  Type `flac' for details.

01 Grendel.wav: wrote 104998973 bytes, ratio=0,575
PS D:\Users\giaco\Desktop\flac-1.3.2-win\win64>
```

Tra le componenti principali che cambiano tra i vari livelli:

la dimensione dei blocchi utilizzati, l'utilizzo del mid size (-m) o adaptive mid size (-M) e l'utilizzo della codifica lineare predittiva rispetto alla codifica lineare con predittori fissi (che migliora la velocita di

encoding, a discapito della dimensione finale), per questo esempio è stata usata la codifica di default (5 quindi con blocchi di 4096 bytes, mid size e con codifica lineare predittiva).

Per la creazione dei file mp3 è stato utilizzato LAME, un encoder open source per la codifica e decodifica dei file mp3.
LAME permette la codifica a bitrate fisso o variabile, il tool permette di usare dei preset predefiniti, oltre che a quelli personalizzati.
Oltre al bitrate, è possibile scegliere la qualità degli algoritmi riguardanti la psicoacustica utilizzando un flag che va da -q0 a -q9, dove q0 è la massima qualità, -q9 è la più veloce, di default il flag è impostato a 3.

Per scegliere il bitrate costante bisogna usare il flag -bxyz dove a posto di xyz va inserito il bitrate che si vuole avere (massimo 320)

```
PS C:\Users\giaco> D:\Users\giaco\Desktop\lame3.100-64\lame.exe --preset insane "D:\Users\giaco\Desktop\lame3.100-64\01
Grendel.wav"
LAME 3.100 64bits (http://lame.sf.net)
CPU features: SSE (ASM used), SSE2
Using polyphase lowpass filter, transition band: 20094 Hz - 20627 Hz
Encoding D:\Users\giaco\Desktop\lame3.100-64\01 Grendel.wav
     to D:\Users\giaco\Desktop\lame3.100-64\01 Grendel.mp3
Encoding as 44.1 kHz j-stereo MPEG-1 Layer III (4.4x) 320 kbps qval=3
    Frame          |  CPU time/estim | REAL time/estim | play/CPU |    ETA
 39635/39635 (100%)|    0:18/    0:18|    0:18/    0:18|   56.300x|    0:00
------------------------------------------------------------------------------
   kbps        LR    MS  %     long switch short %
  320.0       98.8   1.2        95.8   2.4   1.9
```

Per il bitrate variabile si hanno due possibili opzioni, ABR e VBR

ABR setta il bitrate medio, in realtà non sempre il file finale raggiunge quel bitrate, più il bitrate è alto maggiore è la qualità del file, si utilizzata il flag –abr seguito dal bitrate.
VBR setta la qualità del file e elabora il bitrate medio di conseguenza, l'utente a decide un grado di qualità che va da 0 (migliore, maggior spazio) a 9999 (peggiore, minor spazio), ma si occuperà LAME di scegliere il bitrate in base al brano, si utilizza il flag -V seguito dal livello della qualità.

Per il mio file audio ho utilizzato i seguenti valori:

Costanti:

lame -b128 "fileaudio": per avere un brano a 128 kbps con q3
lame –preset insane "fileaudio": corrisponde a -b320kbps con q3

Variabili:

lame –preset 320 "fileaudio": corrisponde a --abr320 con q3
qui di seguito una tabella riassuntiva dei miei file audio dopo le conversioni

	WAV	MP3 128 Costante	MP3 abs 320	MP3 Preset insano	FLAC preset -5
Dimensione (MiB)	36,1 (100%)	3,27 (9%)	7,61 (21%)	8,19 (22,7%)	20,3 (56,2%)
Bitrate (kb/s)	1411	128	297	320	792
Modalità Bitrate	Costante	Costante	Variabile	Costante	Variabile
Compressione	//	Con Perdita	Con Perdita	Con Perdita	Senza perdita

Una volta ottenuto i file ho riconvertito i file audio in formato lossy in wav per fare il confrontare il file ottenuto rispetto all'originale.
Su audacity ho eseguito le seguenti procedure per tutti i file (flac e wave ottenuti dagli mp3):

1. Inversione della traccia: con questo passaggio inverto la fase della mia traccia convertita e per poterla confrontare con l'originale, così facendo i punti in comune nello stesso istante di tempo vengono annullati.
2. Mixaggio delle tracce: ho mixato e creato una nuova traccia, che mostrerà le differenze tra le due tracce

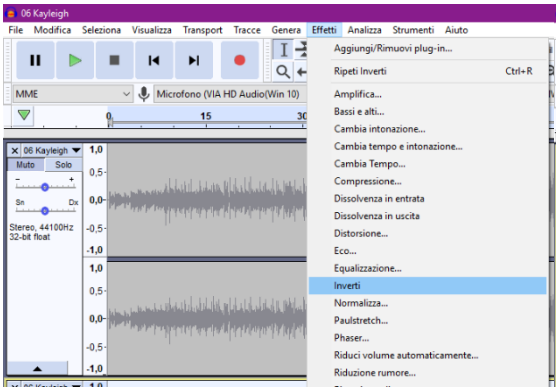


Figura 1-Fase1

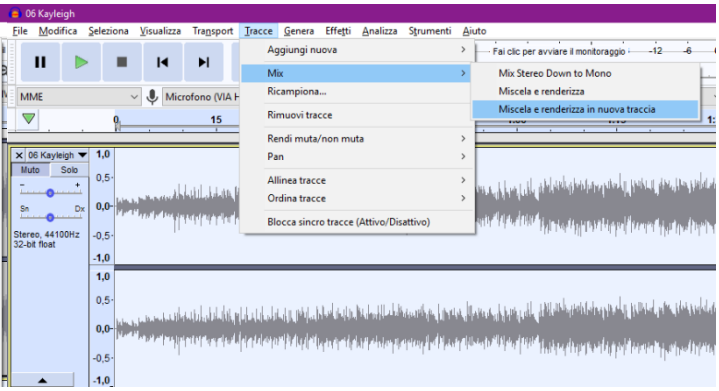


Figura 2-fase2

Una volta analizzate le tracce con audacity, ho provveduto a calcolare PSNR e MSE con Matlab.
Prima di tutto ho importato le tracce nel mio workspace con la funzione:

[y,fs]=audioread("Nome_Traccia_con_directory")

Questa funzione darà luogo a due variabili fs, ovvero la frequenza di campionamento (nel caso di un brano CD 44100Hz), mentre y sarà una matrice in dove nelle il numero di colonne sono i miei canali audio (nel caso stereo 2 colonne) mentre le righe corrispondono ai campioni dei mio file audio per singolo canale.

Per ascoltare il brano importato (con la possibilità di fermarlo) basta creare l'oggetto player in questa maniera:

player=audioplayer(y,fs)

per poi comandarlo con i comandi "play(player)", "stop(player)", "pause(player)" e resume(player).

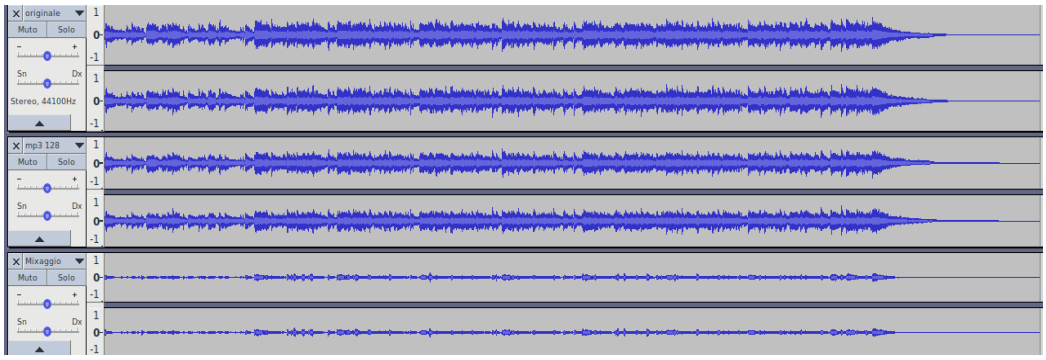
il codice utilizzato su matlab è il seguente:

```
% test di controllo delle dimensioni della mia matrice
[c1x,c1y]=size(y_originale);
[c2x,c2y]=size(y_da confrontare);
if c1x ~= c2x && c1y~=c2y
       disp('dimensioni differenti');
       exit();
else %codice per calcolo
       MSE = sum(sum((y_originale-y_confrontare).^2)/(c1x*c1y));
       S=65535;
       PSNR=-10*log10(MSE/S.^2);
       disp(['mse=' num2str(MSE) ' PSNR= ' num2str(PSNR)]); %num2str converte il
                                  %dato numerico in un array di caratteri per la stampa
end
```

3. Risultati Ottenuti

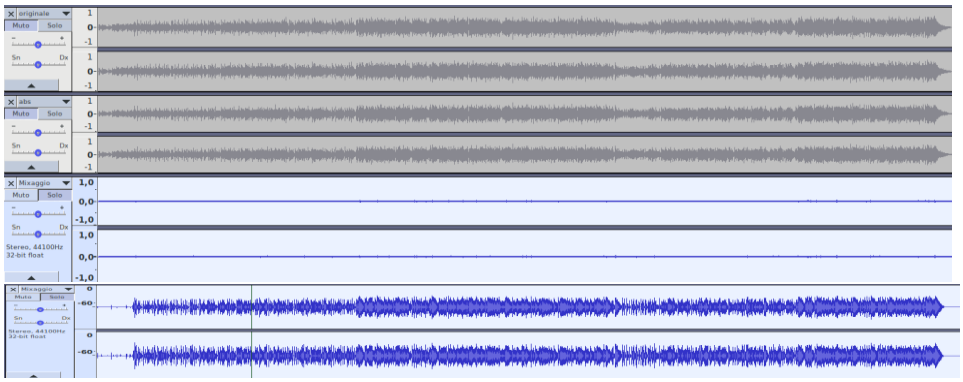
Nota: il livello d'ascolto è prettamente personale è cambia in base al dispositivo utilizzato, le considerazioni audio sono state fatte utilizzando un impianto 5.0 (no sub) con amplificatore Onkyo TX-DS575

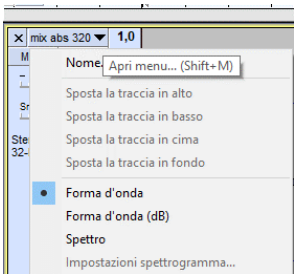
Qui di seguito riporto i risultati ottenuti su audacity
La prima traccia è mp3 128 kbps a bitrate costante



La terza traccia è il mix delle prime due, ascoltandola si evince una perdita audio non indifferente (nella traccia rimangono pure residui di parlato).

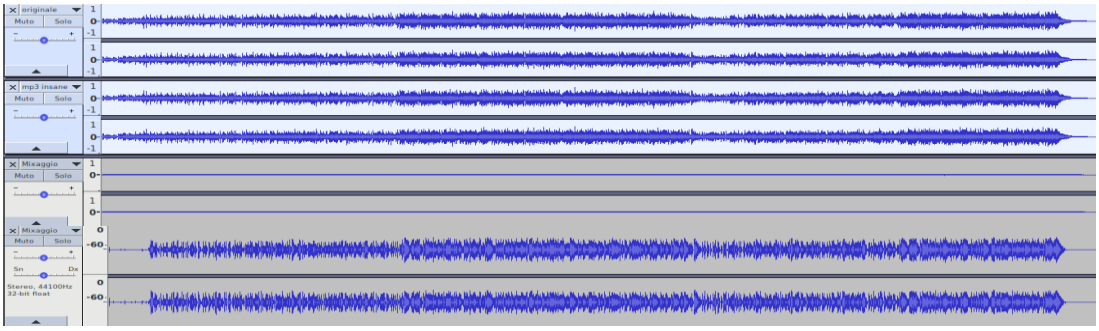
Nella seconda traccia abs320, graficamente la perdita sembra nulla, ma impostando il grafico da forma d'onda a forma d'onda (db), come da immagine, si nota comunque una perdita:





La traccia presenta dei fruscii dopo i primi 10 secondi (molto lenti), le perdite di informazione si accentuano dopo il primo minuto (soprattutto la batteria e parte del parlato).

Nella terza traccia mp3 insane (quindi a 320 kbps, qualità in genere consigliata per l'archiviazione mp3, nonché il massimo bitrate raggiungibile:



Il risultato è molto simile al precedente, non si nota a livello audio un enorme differenza, nonostante il bitrate e le dimensioni maggiori.

Per quanto riguarda il flac invece, sia con la versione convertita, che utilizzando lo stesso formato (senza riconversione) si nota una forma d'onda nulla, nonché una traccia perfettamente silenziosa

Il mixaggio in immagine è già impostato su Audacity in forma d'onda (db).

Dopo aver calcolato il PSNR e MSE con Matlab riporto qui di seguito in aggiunta alla tabella precedente

	WAV	MP3 128 Costante	MP3 abs 320	MP3 Preset insano	FLAC preset -5
Dimensione (MiB)	36,1 (100%)	3,27 (9%)	7,61 (21%)	8,19 (22,7%)	20,3 (56,2%)
Bitrate (kb/s)	1411	128	297	320	792
Modalità Bitrate	Costante	Costante	Variabile	Costante	Variabile
Compressione	//	Con Perdita	Con Perdita	Con Perdita	Senza perdita
MSE	0	$1{,}3038\times10^{-4}$	$4.4083*10^{-6}$	$3.6757*10^{-6}$	0

PSNR	∞	135.1774	149.8867	150.6761	∞

Dai dati ottenuti e dai confronti fatti, si arriva alla conclusione che il formato FLAC conserva interamente i dati, riuscendo a comprimere il 43% circa, utilizzando le impostazioni di default del tool (valore che può aumentare aumentando la qualità di codifica a un valore maggiore di 5.
Gli mp3 tendono a occupare meno spazio rispetto a un file con codifica lossless, ma tendono a eliminare dati non sempre inutili, anche se aumentando la qualità, la differenza audio tende sempre più ad assottigliarsi, anche se non si annulla mai del tutto, i valori ottenuti con i test, dimostrano che la codifica abs e cbs (constant bitrate) sono molto simili tra di loro, anche se si nota una leggera vittoria di quest'ultima.

4. Appendice

Abr: average bit rate

Audacity: software per la modifica e l'analisi di file audio, licenza GNU (https://manual.audacityteam.org)

Cbr: constant bit rate

Flac Tool: strumenti ufficiale per il formato flac (https://xiph.org/flac/documentation_tools_flac.html)

Free:ac: programma Open Source (https://www.freac.org)

LAME: software open source per la codifica mp3 (https://svn.code.sf.net/p/lame/svn/trunk/lame/USAGE)

Lossless: senza perdita di informazione

Lossy: con perdita di informazione

MATLAB: ambiente software per il calcolo e l'analisi numerica, licenza commerciale a pagamento (https://it.mathworks.com/products/matlab.html?s_tid=hp_products_matlab), un alternativa (che funziona senza modificare il codice .m presente in questa documentazione) è Octave (https://www.gnu.org/software/octave)

Ripping: estrazione della traccia audio

VBR: Variable bit rate