

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

SIGNÁLY A SYSTÉMY

2019/2020

PROJEKT

1. Úkol

Mé nahrávky můžete použít způsobem (a), (b), (c)

Název	Délka ve vzorcích	Délka v sekundách
sa1.wav	86000	5.38
sa2.wav	60080	3.75
si1576.wav	31760	1.99
si2206.wav	97760	6.11
si946.wav	59600	3.73
sx136.wav	80720	5.04
sx226.wav	63440	3.96
sx316.wav	44720	2.79
sx406.wav	74000	4.62
sx46.wav	44240	2.77

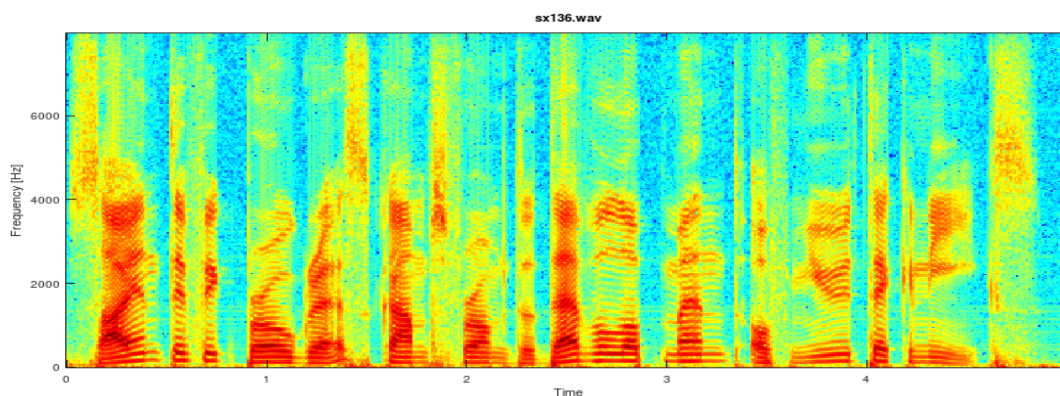
2. Úkol

Název	Délka ve vzorcích	Délka v sekundách
q1.wav	14304	0.89
q2.wav	16098	1.01

3. Úkol

viz. funkce Spectrogram.m (či funkce v „**program.py**“)

spuštění (v Octave) např. „Spectrogram('..sentences/sx136.wav')“



4. Úkol

Pro výpočet parametrů jsem zvolil postup, který naleznete v souboru „CalculateParameters.m“ (popř. v **program.py**). Postupoval jsem na základě zadání a sice $F = A * P$. Matici A jsem tvořil následovně:

1. vytvořil jsem si matici o velikosti A o velikosti 16 x 256, kterou jsem naplnil nulami
2. Pomocí *for* cyklů a proměnné „step“ jsem postupně naplnil první řádek jedničkami od 1 do 16, poté k proměnné step přičetl 16 a následoval druhý řádek od indexu 17 do 32...
3. V posledním kroku jsem obě matice mezi sebou vynásobil, díky tomu, že kromě míst, kde je v matici A hodnota 1 byly všude jinde 0, sečetlo se mi vždy 16 původních čísel do jednoho.

(popsané řešení je pro indexaci Octave)

5. Úkol

viz. soubory QueryScore.m a QueryPom.m (či v „program.py“)

```
function retval = QueryScore (queryM, sentenceM)

total_steps = columns(sentenceM)-columns(queryM); # maximální počet kroků - nesmím jít mimo hranice
resultVect = zeros(1,columns(sentenceM)); # vektor pro výsledky korelace

for i = 1:total_steps # i = pp ze zadání
    pomKorelace=QueryPom(queryM, sentenceM, i); # pro každé "pp" počítám korelaci pomocí fce corr
    resultVect(i)=pomKorelace; # výsledky ukládám
endfor

retval = resultVect;
endfunction

function retval = QueryPom (queryM, sentenceM, i)

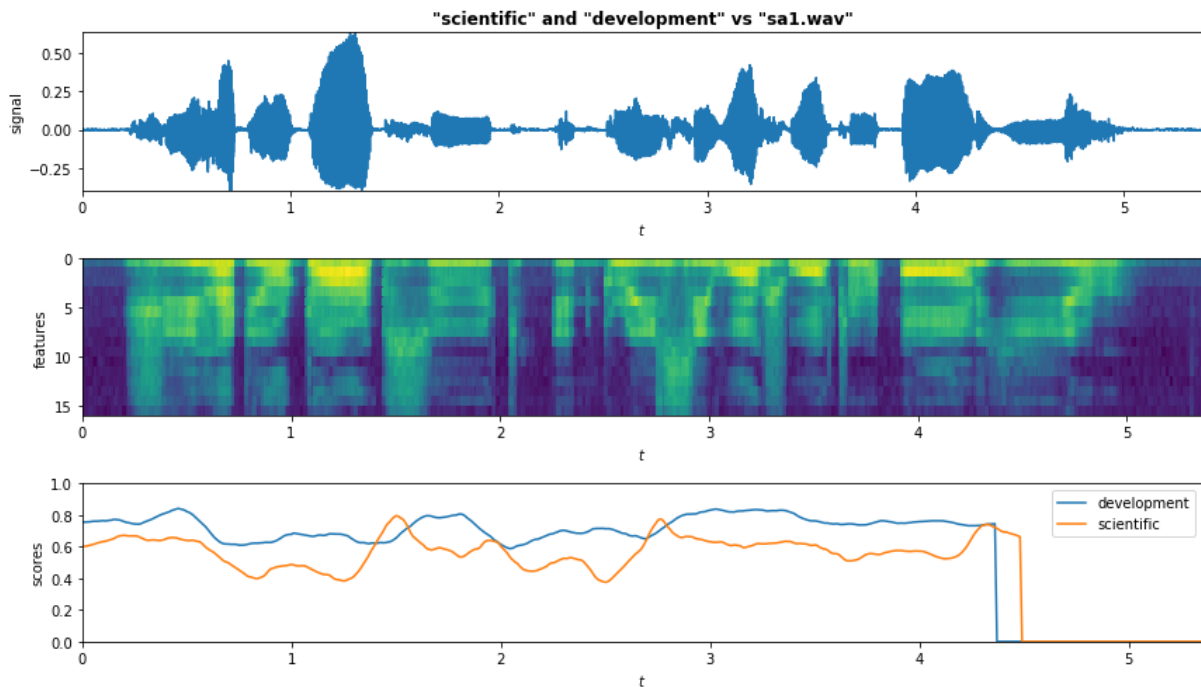
korelaceVysledek=0; # výsledek korelace pro aktuální "pp"
pom=0; # pomocná proměnná pro výpočet korelaci

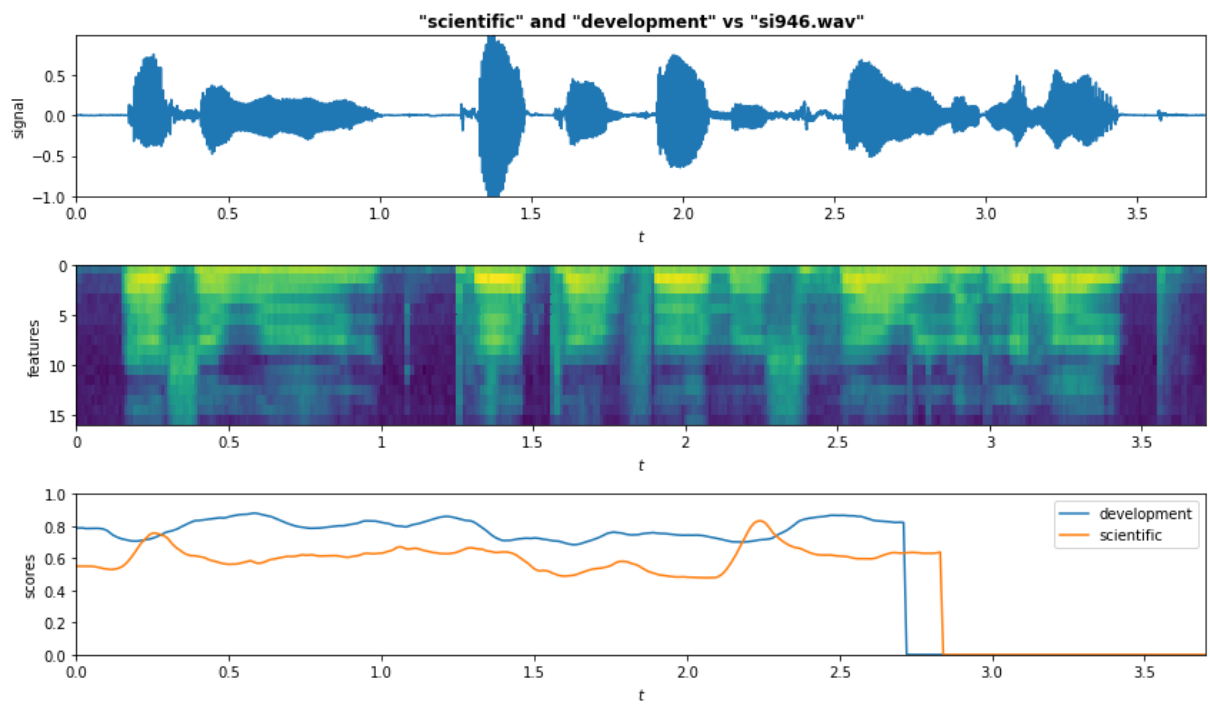
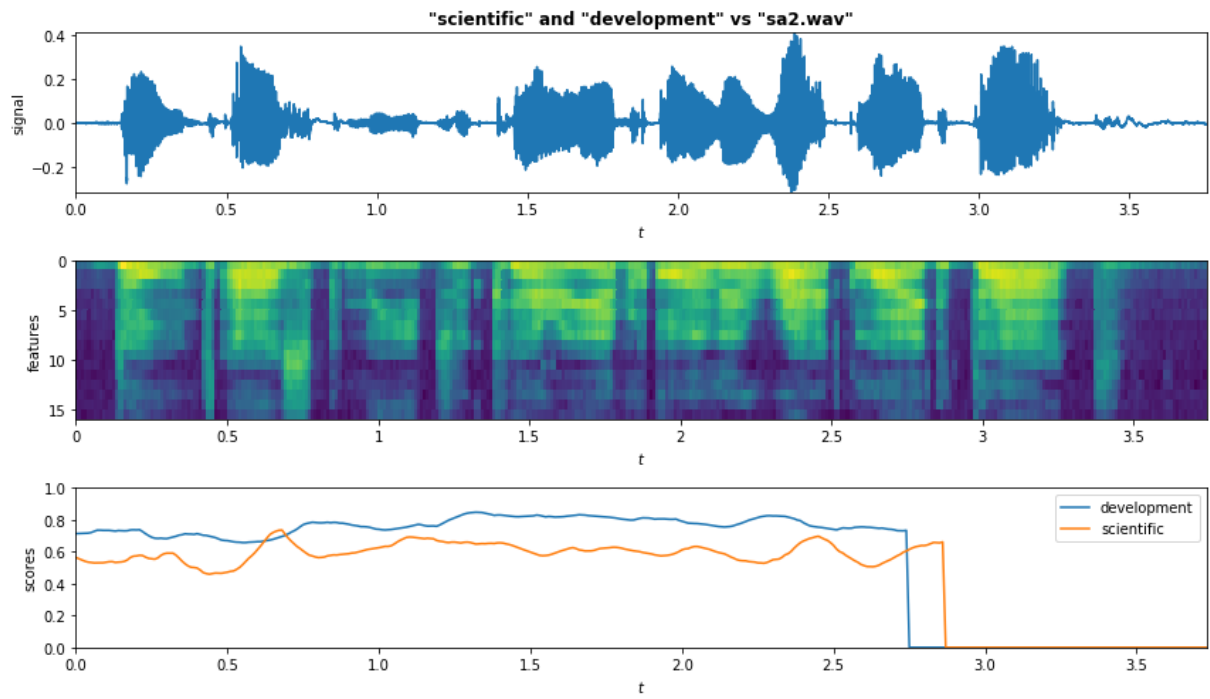
for sloupec = 1:columns(queryM)
    pom=corr(queryM(1:16, [sloupec]), sentenceM(1:16, [sloupec + i])); # výpočet korelace pomocí funkce "corr"
    korelaceVysledek += pom;
endfor

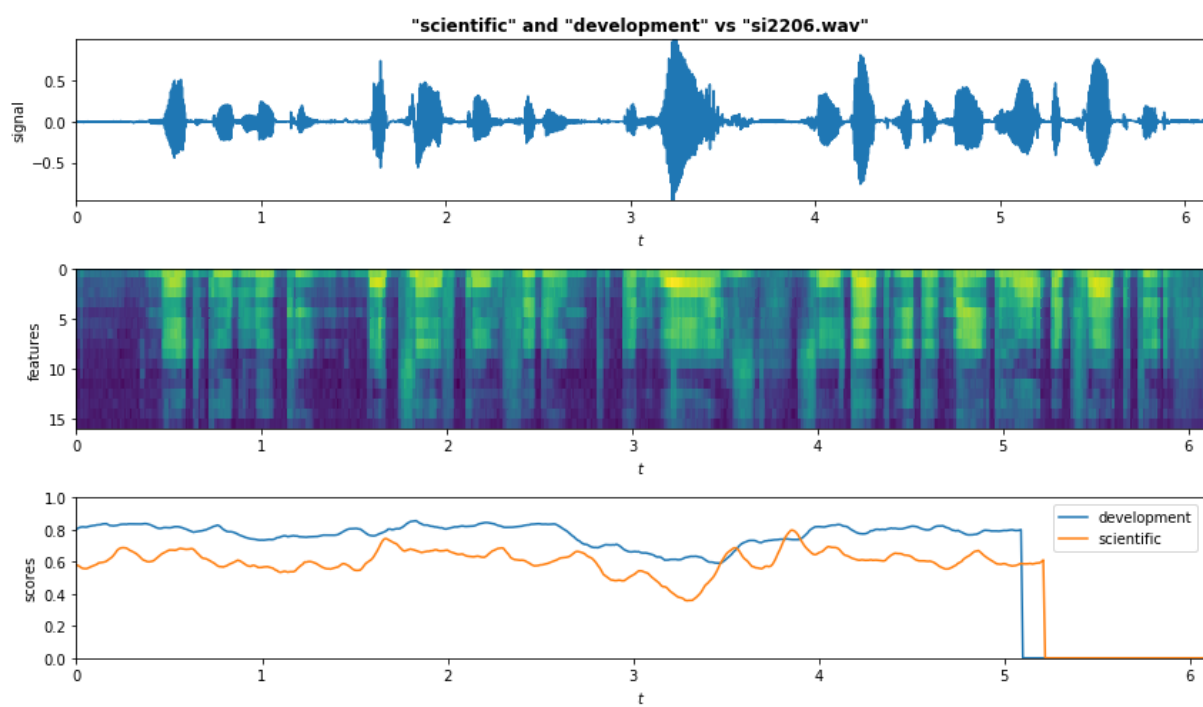
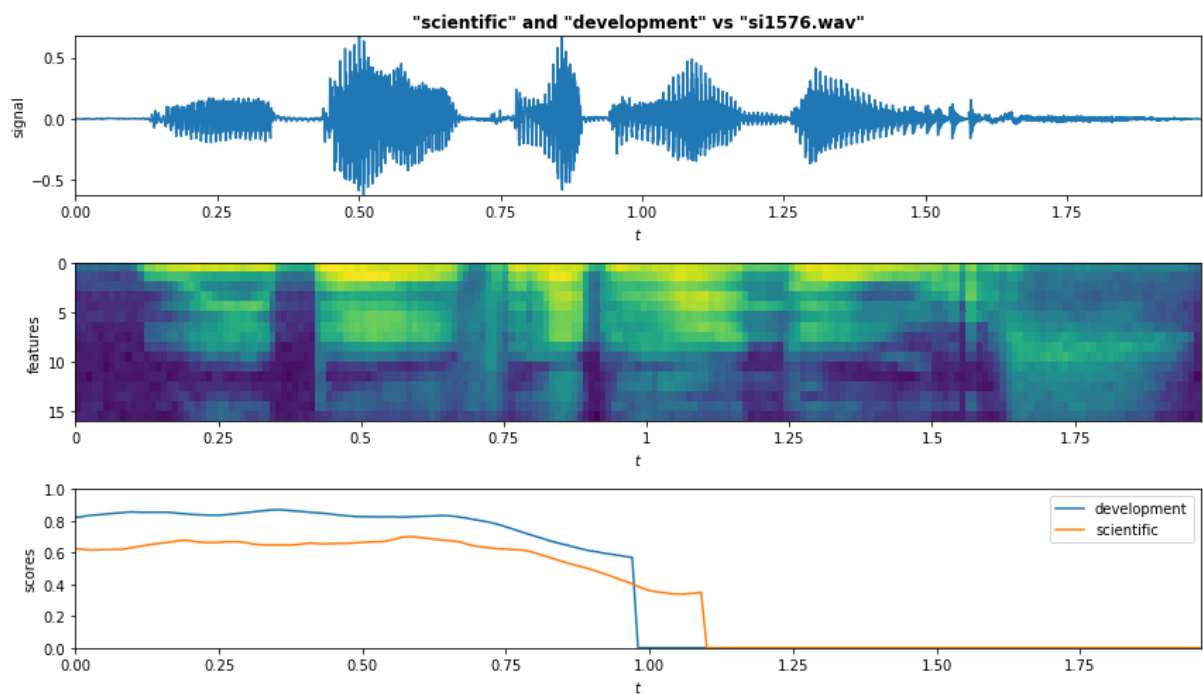
retval=korelaceVysledek / columns(queryM); # výsledek chci mezi -1 a 1, tudíž musím hodnotu normalizovat
endfunction
```

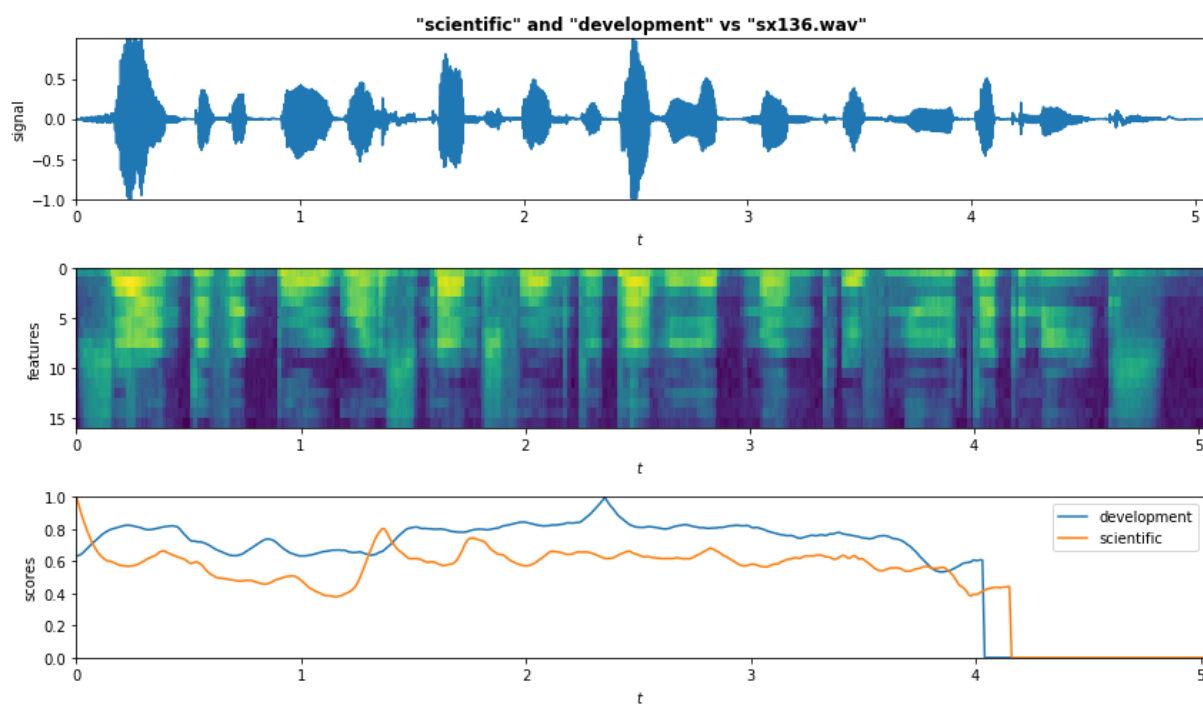
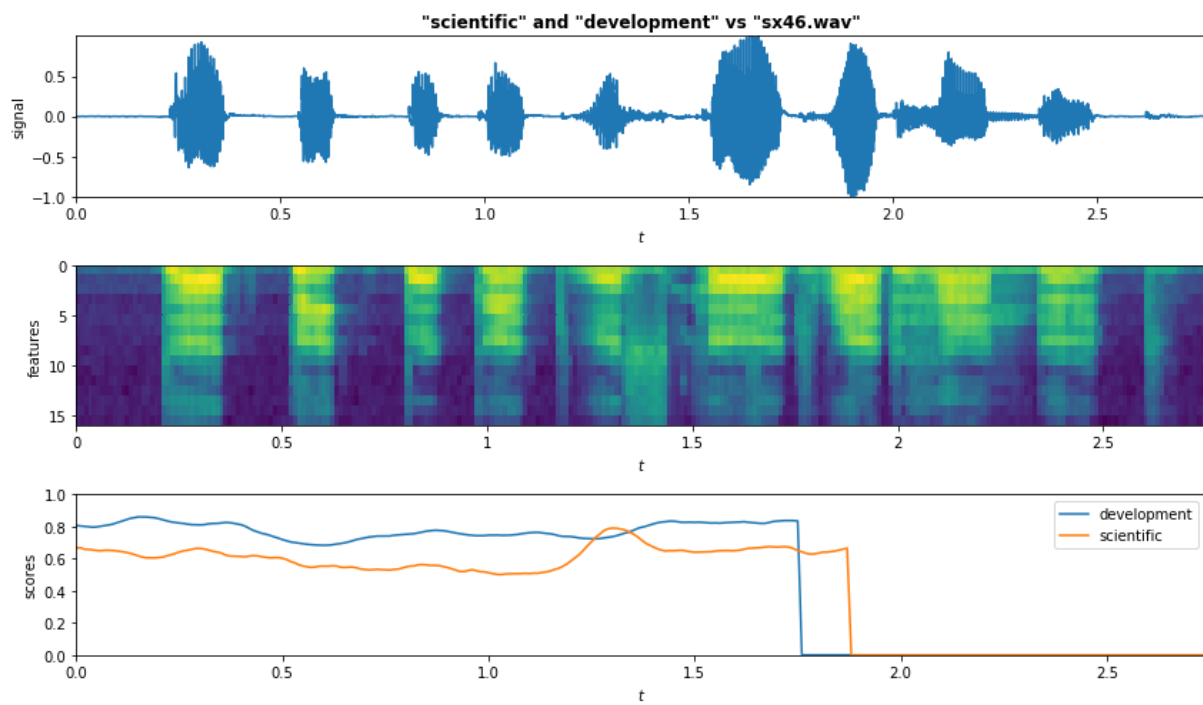
6. Úkol

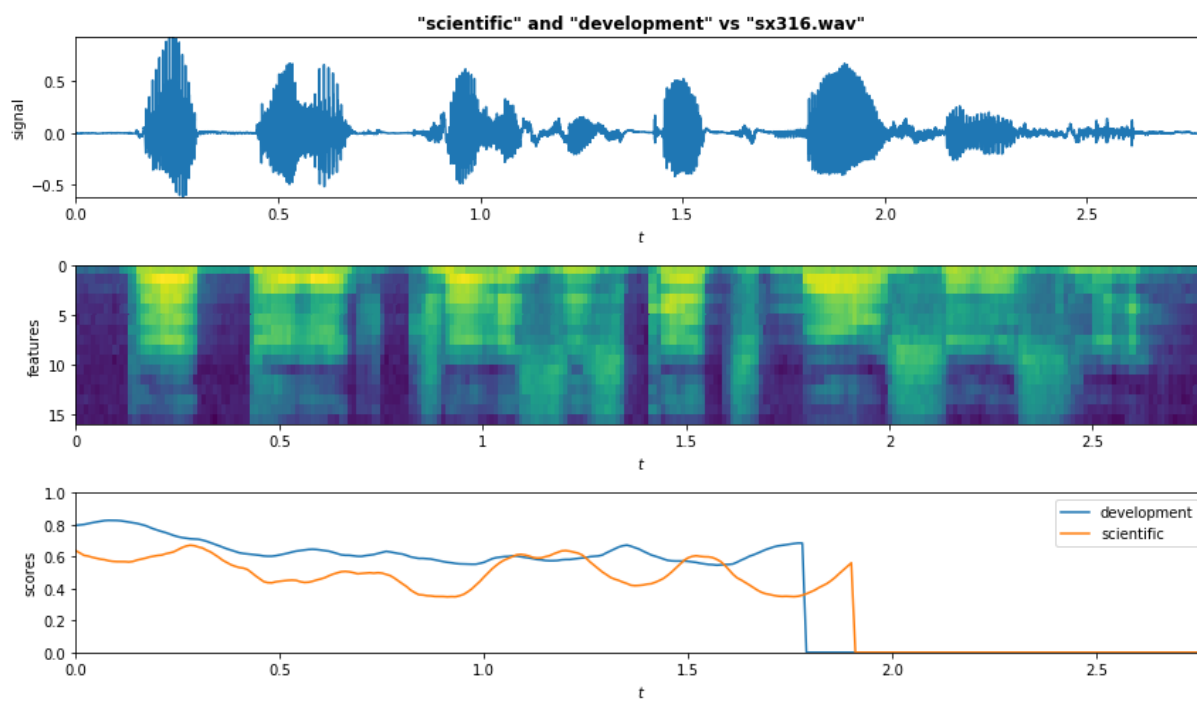
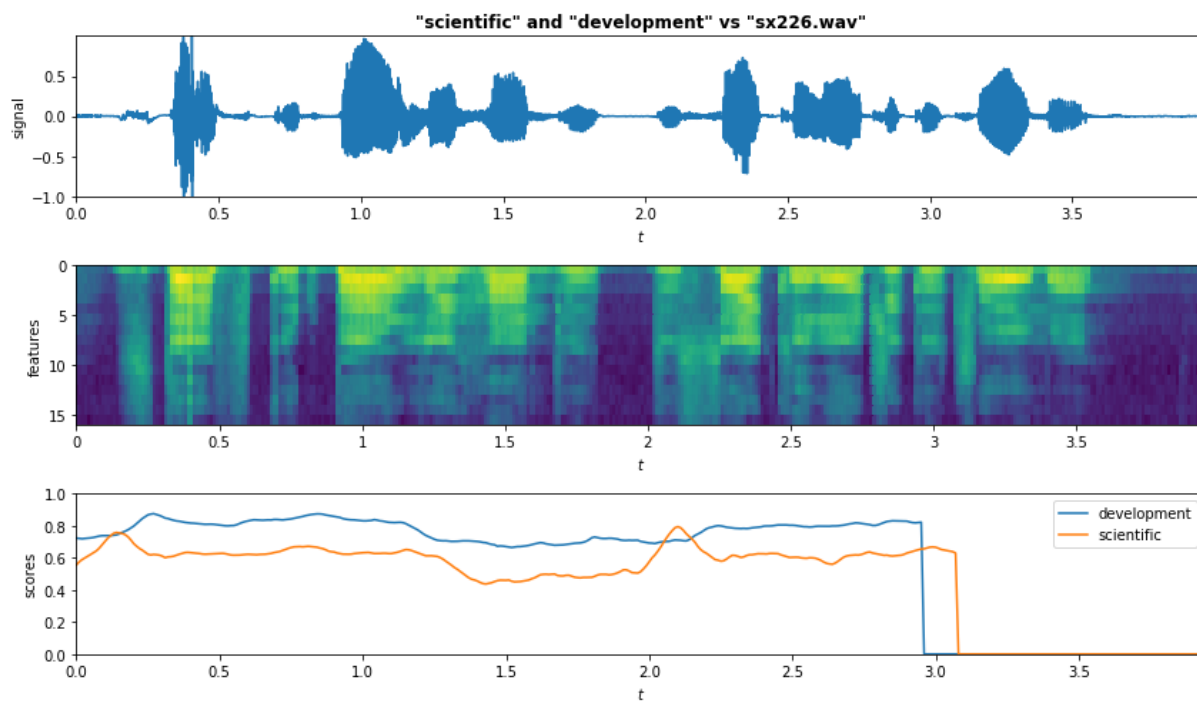
10 obrázků (časová osa X je v sekundách), výsledky z „program.py“

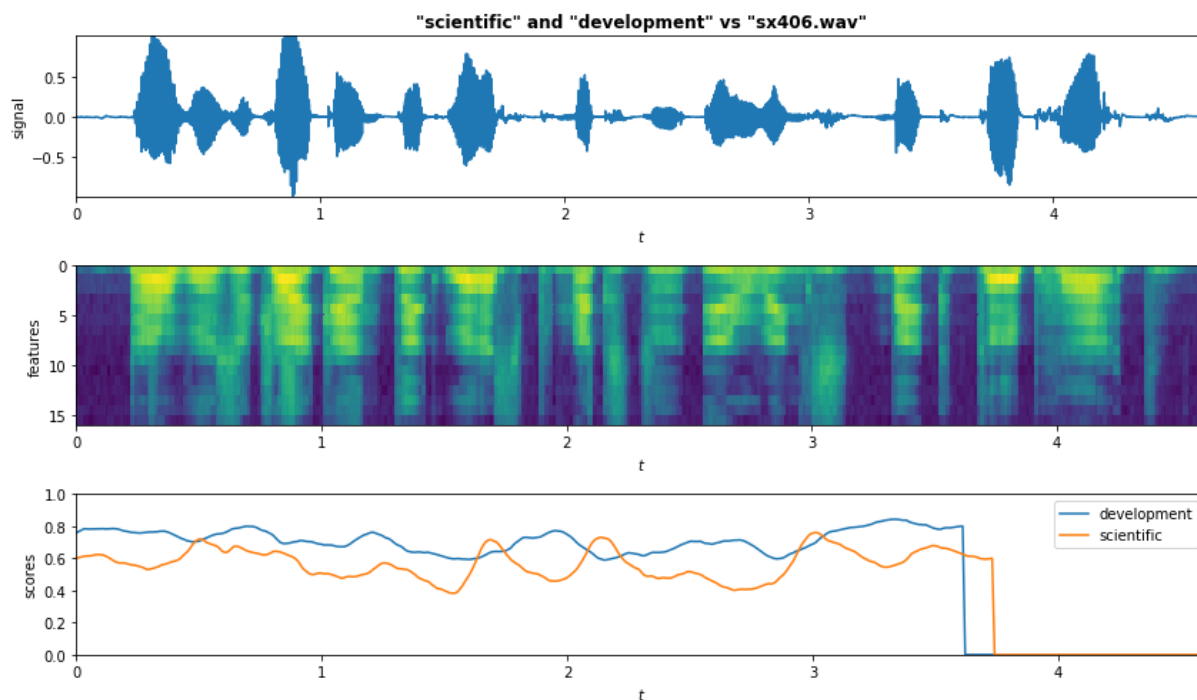












7. Úkol

Práh (threshold) pro konkrétní případ mých nahrávek dle výsledků sledování bude roven 0,9 tzn. (skóre $\geq 0,9$ = hit) pro „q1.wav“ i pro „q2.wav“.

8. Úkol

Výsledky „hits“ jsou ve složce „hits“

Název	Klíčové slovo Q1 (pokud ANO vzorky od – do)	Klíčové slovo Q2 (pokud ANO vzorky od – do)
sa1.wav	NE	NE
sa2.wav	NE	NE
si1576.wav	NE	NE
si2206.wav	NE	NE
si946.wav	NE	NE
sx136.wav	ANO, 200-14304	ANO, 37000-53098
sx226.wav	NE	NE
sx316.wav	NE	NE
sx406.wav	NE	NE
sx46.wav	NE	NE

9. Úkol

Můj detektor podle mého víceméně funguje, ovšem záleží na kvalitě nahrávky. Detektor jsem prvně implementoval v Octave, ovšem původní nahrávky mi ukazovaly celkem nesmyslné výsledky – pro každou kombinaci byly hodnoty skóre $>0,85$... Byl jsem přesvědčen o tom, že programově můj postup musí být v souladu se zadáním a že tedy bude problém v některé z funkcí (zejména jsem měl podezření u funkce „corr“ pro výpočet Pearsonova korelačního koeficientu), načež jsem tedy celý program přepsal i do Pythonu, při přepisu jsem kvůli rozdílné indexaci udělal chybu, kterou jsem

následně opravil, ale výsledky byly stejné, jako v Octave. Napadlo mě tedy zkusit jiné nahrávky, s kvalitnějším mikrofonom... A najednou byly výsledky „správné“, takže mám nyní projekt implementovaný (prakticky kompletní) ve dvou jazycích – program.py (Python), *.m (dílčí funkce v Octave).

Detektor se zlepšit určitě dá, jak programově – rozdělení na více „dílků“ apod. tak i kvalitou nahrávek (studiový mikrofón?).

Pozn. „**program.py**“ obsahuje kompletní projekt, v Octave chybí vykreslování „kompletních“ obrázků atd.