Travail d'analyse d'EEG

Preparation

```
> restart; with(DiscreteTransforms) : with(plots) : with(Statistics) : with(LinearAlgebra) :
> MedianPass := proc(entre)
  local dimm, j;
  dimm := Dimension(entre) :
  for j to dimm do
  Vector([entre[1], seq(sort([seq(entre[i], i = j..j + 2)])[2], j = 1..dimm - 2),
      entre[dimm]], datatype = float[8])
  end do:
  end proc:
> VectorDWT := proc(V, iters, w1, w2, $)
      local i, s, temp1, temp2;
   s := LinearAlgebra:-Dimensions(V);
   if modp(s, 2^iters) <> 0 then
     error "the length of V is not divisible by enough powers of 2"
   end if;
   temp1 := Vector(V);
   temp2 := Vector(V);
   for i to iters do
     DiscreteWaveletTransform(s/2^(i - 1), temp1, w2, w1, storagetype = singlearray,
     temp2);
     ArrayTools:-Copy(temp2, temp1) ;
   end do;
   return temp1
  end proc:
> InverseVectorDWT := proc(V, iters, w1, w2, $)
      local s, i, temp1, temp2;
   s := LinearAlgebra:-Dimensions(V);
   if modp(s, 2^iters) <> 0 then
     error "the length of V is not divisible by enough powers of 2"
   end if;
   temp1 := Vector(V);
   temp2 := Vector(V);
   for i from iters by -1 to 1 do
     InverseDiscreteWaveletTransform(s/2^(i - 1), temp1, w2, w1, storagetype
      = singlearray, temp2);
     ArrayTools:-Copy(temp2, temp1);
   end do;
   return temp1
  end proc:
> EnergySpectrum := proc(entre, nbrpas, pastemps)
```

```
local SignalBrute, n, SignalBrutecut, times, Centrage, Signalcentr, SDw1, SDw2, SDT,
    SDR, signalbas, Signalnet, wavelets, Gra, ES ;
SignalBrute := Vector( [entre], datatype = float[8]) :
n := Dimension(SignalBrute) :
SignalBrutecut := Vector( [entre[1 ..nbrpas ]], datatype = float[8]) :
times := Vector( [seq( i/pastemps, i = 0 ..nbrpas − 1 )], datatype = float[8] ) :
Centrage := Mean(SignalBrutecut) :
Signalcentr := SignalBrutecut − Vector( [seq(eval(Centrage), i = 1 ..nbrpas )], datatype
    = float[8]) :
SDw1, SDw2 := WaveletCoefficients(Coiflet, 4) :
SDT := VectorDWT(Signalcentr, 3, SDw1, SDw2) :
SDR := Vector( [seq( SDT[i], i = 1 .. nbrpas/2 ), seq( 0, i = 1 .. nbrpas/2 )], datatype
    = float[8] ) :
signalbas := InverseVectorDWT(SDR, 3, SDw1, SDw2) :
Signalnet := MedianPass(signalbas) :
wavelets := DiscreteWaveletTransform(Signalnet, Daubechies, 4) :
Gra := plots[listplot](times, Signalnet, title = "EEG sous filtre passe-bas et median") :
ES := evalf( log10( add( abs( wavelets[1][i]·wavelets[2][i]^2 ), i = 1 .. nbrpas/2 ) ) ) :
printf("Log Energy Spectrum : %f\n", ES);
print(plots[display]({Gra}));
 end proc:
```

```
>
```

Analyse

Z005

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
    2013/MAP-6014/Info/Application/Z/Z005.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

Z009

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
    2013/MAP-6014/Info/Application/Z/Z009.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

Z011

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/Z/Z011.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

Z015

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/Z/Z015.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

S004

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/S/S004.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

S007

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/S/S007.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

S010

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/S/S010.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

O010

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/O/O010.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

O011

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/O/O011.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

N005

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/N/N005.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

N015

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/N/N015.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

F007

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/F/F007.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

F009

```
> fo
        := fopen("/Users/sebastienbertrand/Documents/Études/Mathématique/Été
      2013/MAP-6014/Info/Application/F/F009.txt", READ, TEXT) :
> data := readdata(fo, integer, 1) :
> fclose(fo)
> EnergySpectrum(data, 512, 173.61)
```

```
>
```