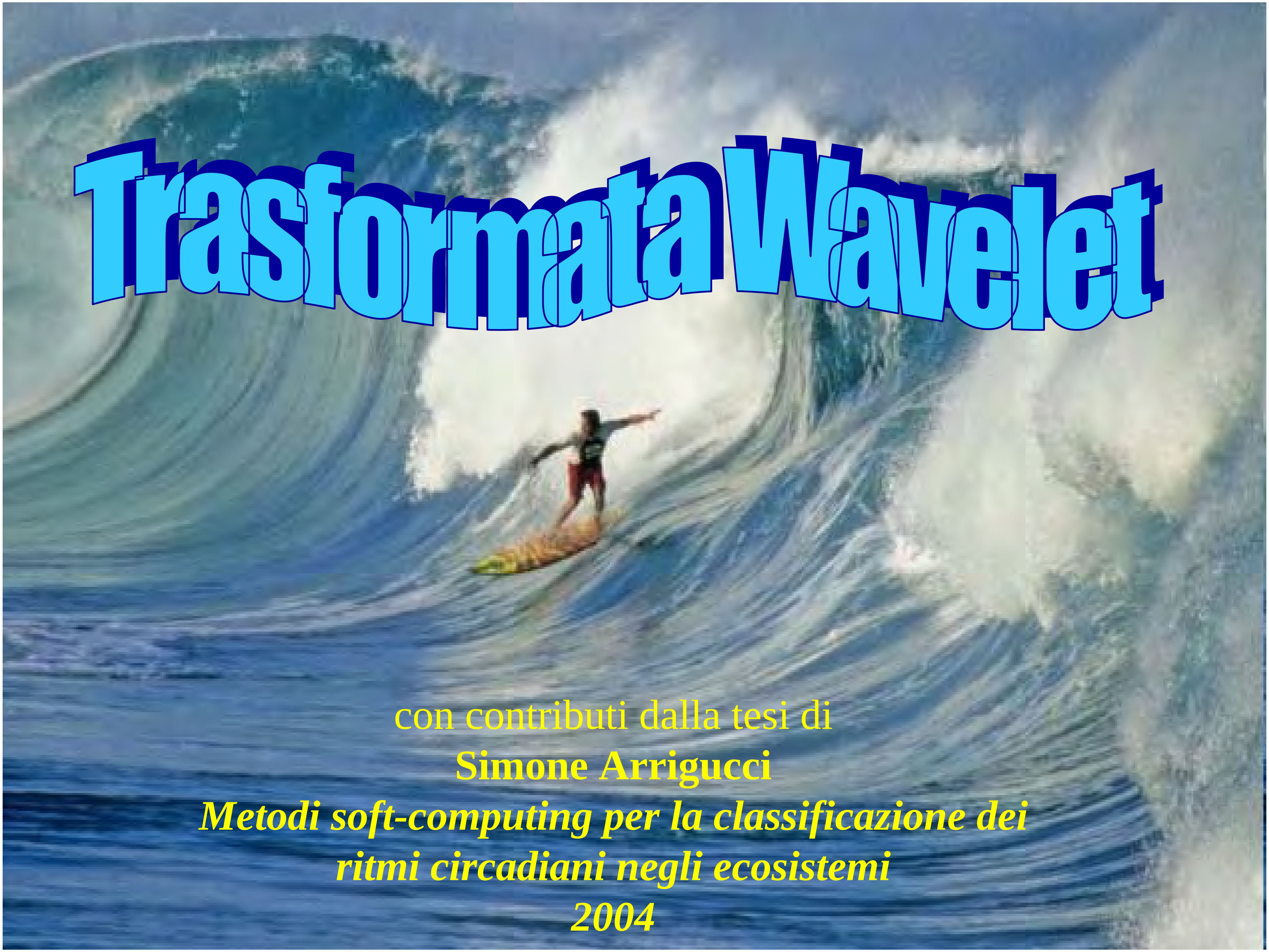


Trasformata Wavelet

con contributi dalla tesi di

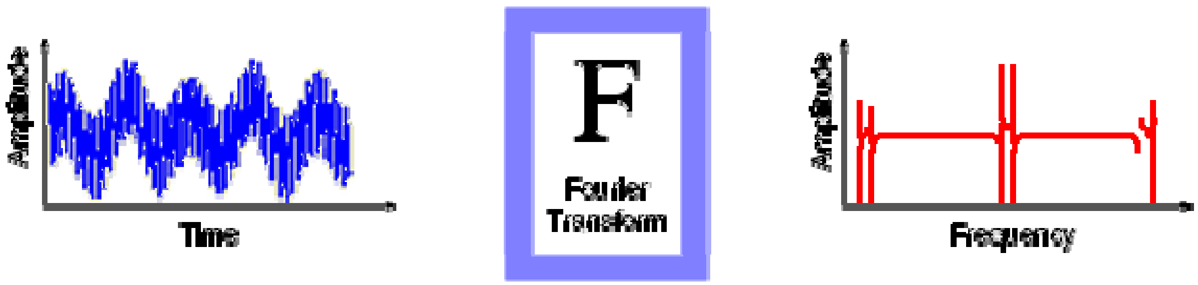
Simone Arrigucci

Metodi soft-computing per la classificazione dei ritmi circadiani negli ecosistemi

2004

Oltre Fourier

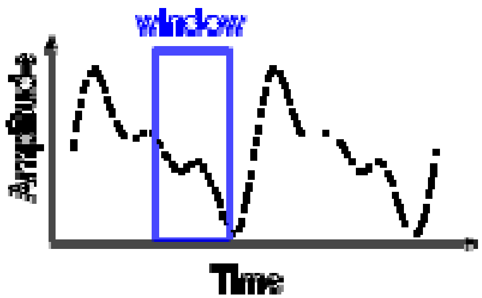
- L'analisi di Fourier è molto utile perché decompone un segnale nelle sue armoniche. Ne rivela perciò il contenuto in frequenza spostando l'analisi dal dominio del tempo a quello della frequenza

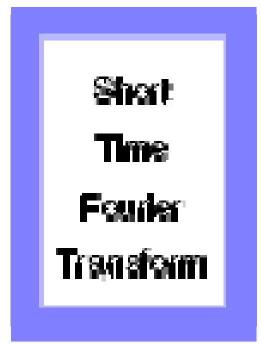


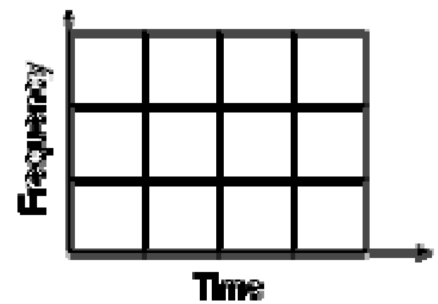
- Perché abbiamo bisogno di altri strumenti?
- Limite di Fourier: passando dal tempo alla frequenza l'informazione "tempo" si perde.
  - Guardando alla F-trasformata non possiamo dire quando un particolare evento è avvenuto
  - ***Perdita dell'informazione temporale***
  - *Suppone il segnale* ***stazionario***

Short-term Fourier Transform (STFT)

- Nel 1946 Gabor adattò la F-transform per considerare solo una porzione del segnale attraverso una “finestra” temporale



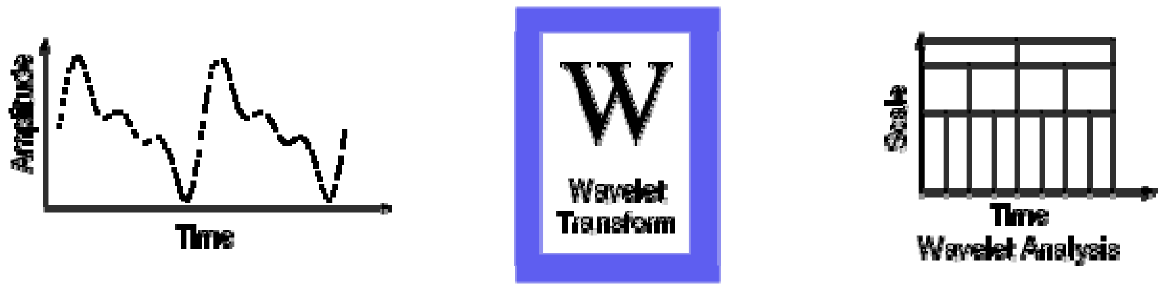




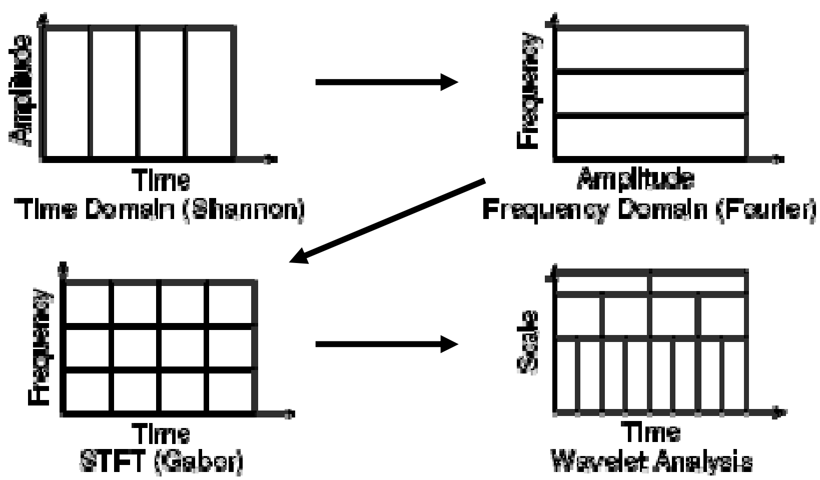
- La STFT è un compromesso fra tempo e frequenza, ma la sua precisione dipende dall’ampiezza della finestra
- Il suo limite è che una volta scelta una dimensione per la finestra, essa vale ***per tutte le frequenze***
- Molti segnali richiedono un approccio più flessibile
  - La finestra dovrebbe adattarsi alle esigenze di scala sia nel tempo che nelle frequenze.

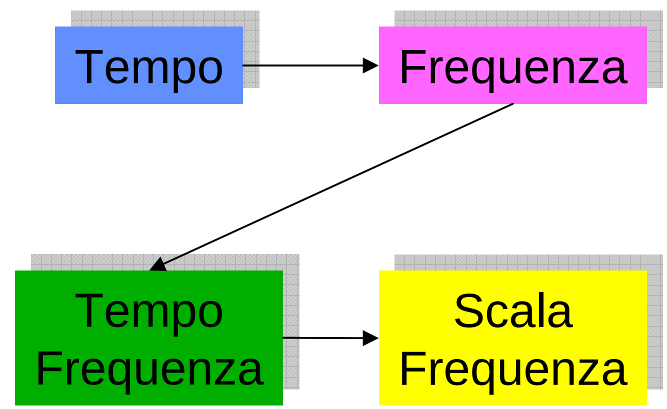
Wavelet Trasform (WT)

- La WT è il successivo passo logico: finestre adattive
  - Intervalli di tempo ***lunghi*** per analizzare le ***basse*** frequenze
  - Intervalli di tempo ***corti*** per analizzare le ***alte*** frequenze



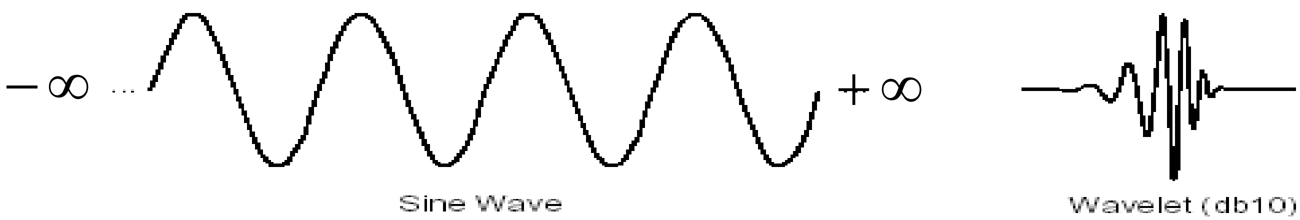
- Storicamente, la progressione nell'analisi è:





L'essenza delle Wavelet

- Come suggerisce il nome una *wavelet* è una *"piccola onda"* che è $\neq 0$ solo in un periodo di tempo limitato.
- E'opposta alla *"grande onda"*, come la *funzione seno*, che esiste da -∞ a +∞ (Fourier)



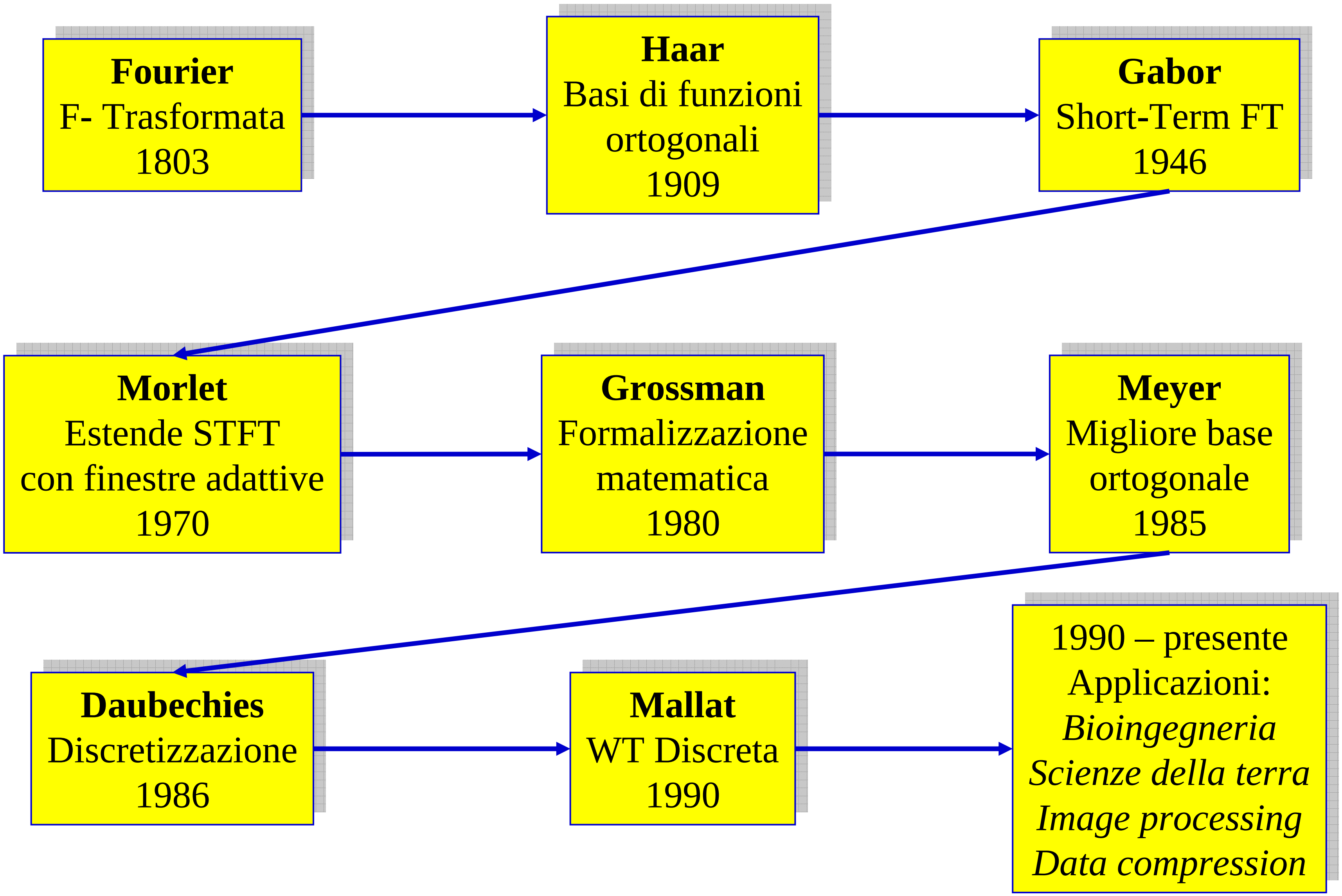
- Una *wavelet* soddisfa a due fondamentali proprietà

$$\int_{-\infty}^{\infty} \psi(u)du = 0 \qquad \int_{-\infty}^{\infty} \psi^2(u)du = 1$$

media nulla — energia finita

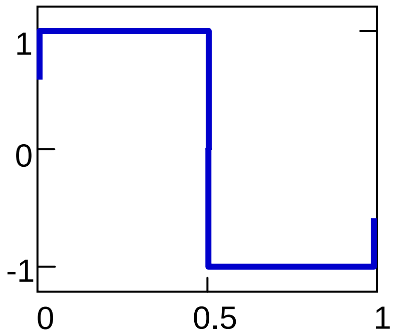
- Una *wavelet* esiste solo in un intervallo finito (-T, T) dove assume sia valori positivi che negativi

Cronologia delle Wavelet

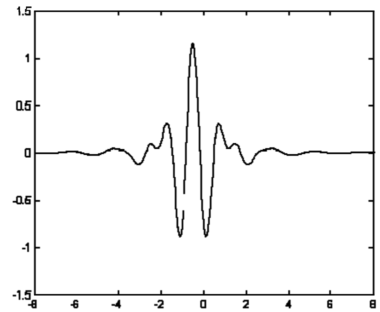


Alcune funzioni wavelet

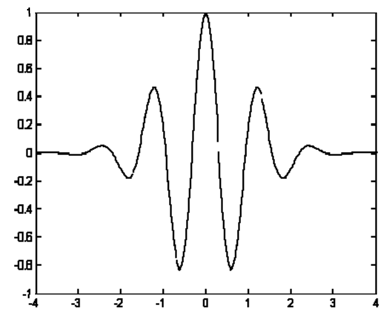
Haar = db1



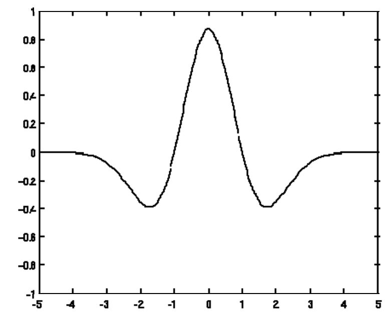
Meyer



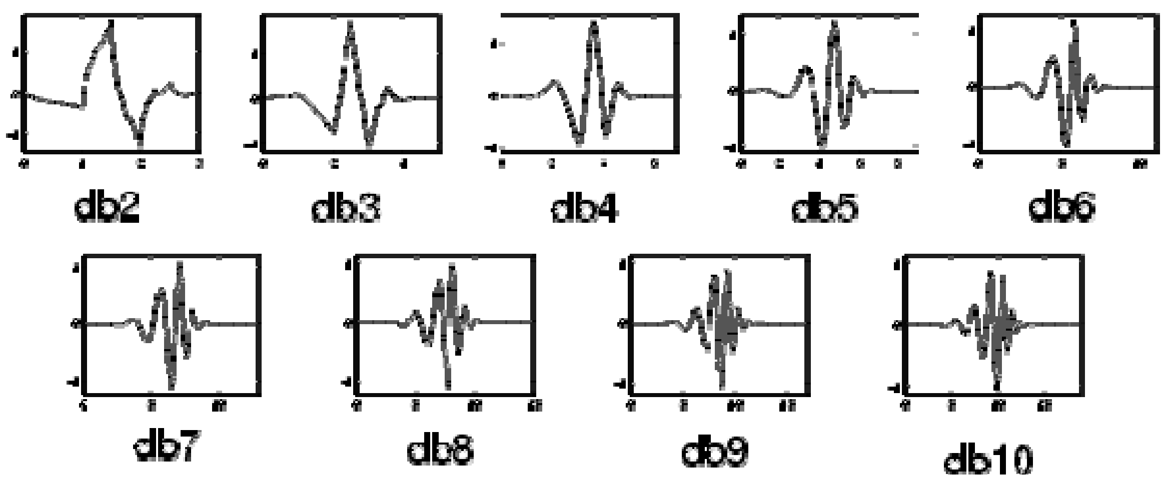
Morlet



Mexican hat



Daubechies



Pagine web
con tutorials
sulle wavelet

http://www.amara.com/IEEEwave/IEEEwavelet.html

http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

Come opera la *Wavelet Continua*

- Si ripete il concetto di approssimazione locale, come per le *splines*, ma su 2 dimensioni (tempo e scala)
- Il segnale viene espresso mediante combinazioni di *wavelets “figlie”*, frutto dello *shifting* e *scaling* della stessa *wavelet “madre”*

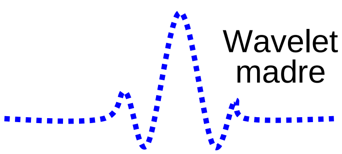
$$C(scale, shift) = \int_{-\infty}^{\infty} s(t) \cdot \psi(scale, shift, t)\, dt$$

- Il vantaggio è di poter adattare le *wavelets* sia in ampiezza che in frequenza in modo adattivo, seguendo le caratteristiche del segnale, attraverso le proprietà di *shifting scaling*

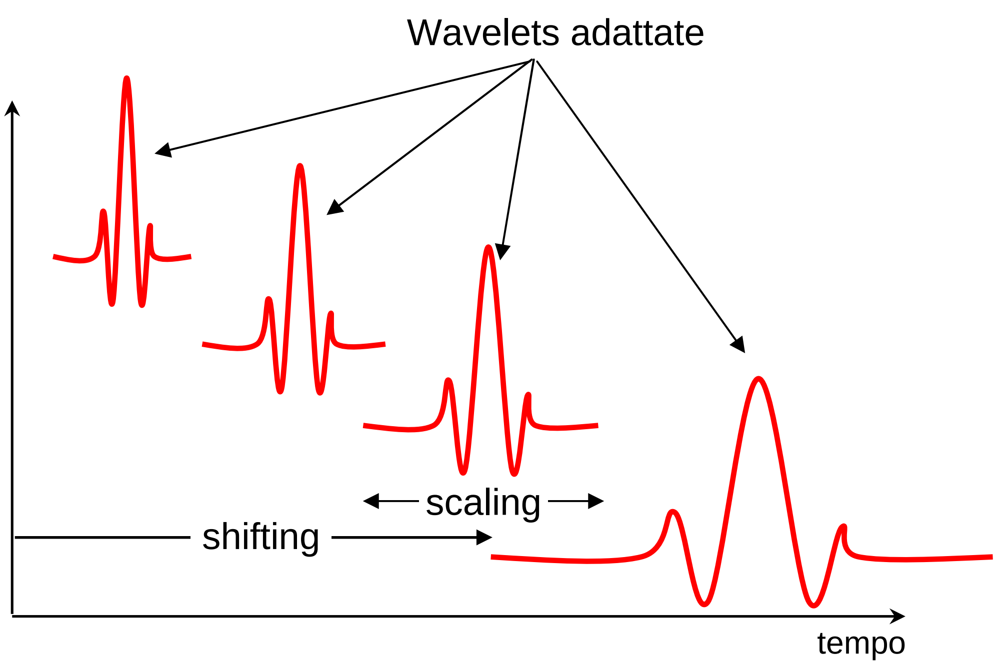
Scaling e *Shifting* delle funzioni Wavelet

☞ Data una generica wavelet $\psi(a,b,t)$, dove *a* e *b* rappresentano i fattori di ***scaling*** e di ***shifting***, la trasformata wavelet continua (CWT) è definita come l'integrale del segnale *s(t)* moltiplicato per la wavelet scalata

$$C(a,b)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} s(t)\cdot\psi\left(\frac{t-b}{a}\right)dt$$

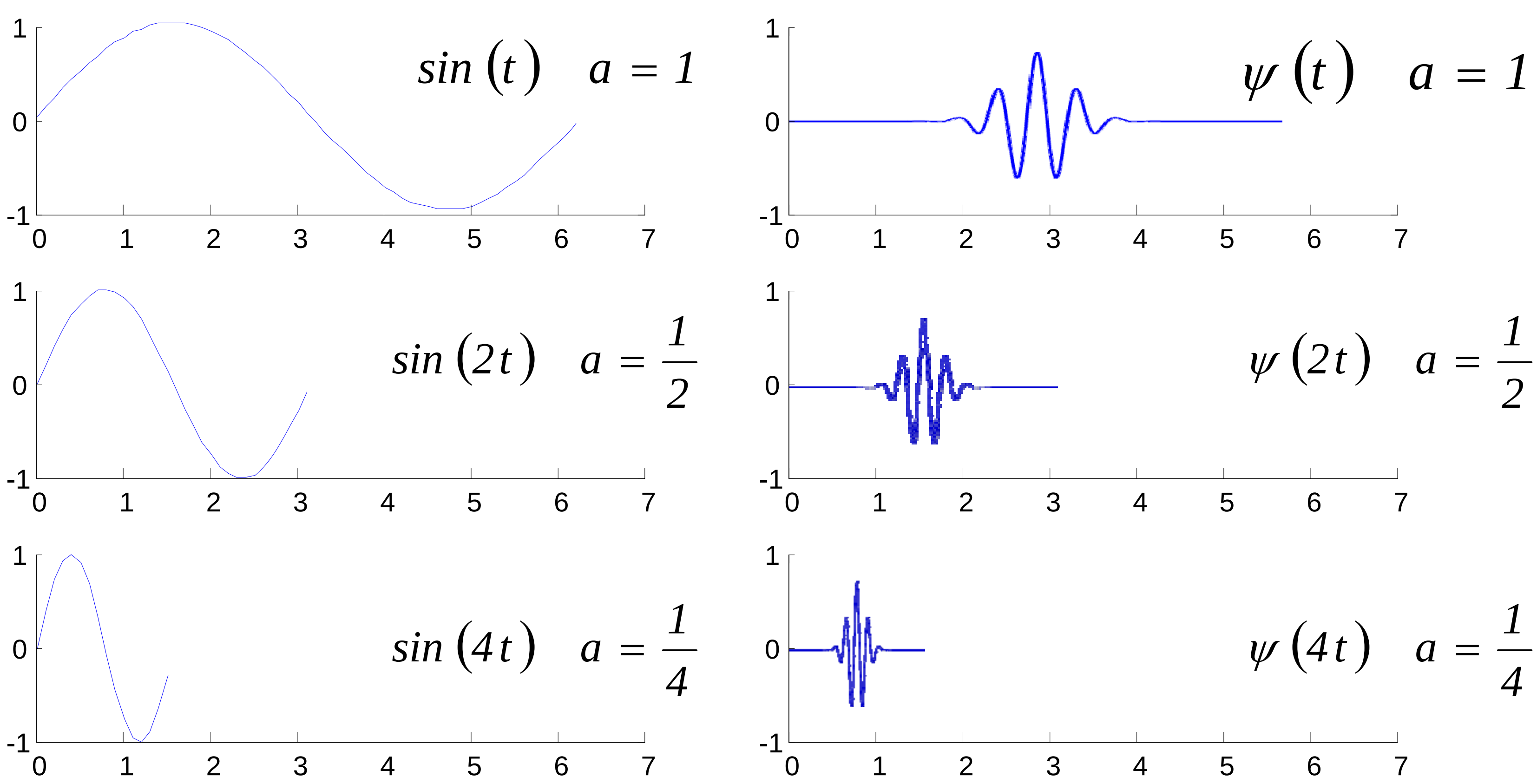


- Se la wavelet viene dilatata *(a >> 1)* contiene prevalentemente ***basse frequenze*** e fornisce un'approssimazione globale del segnale
- Se è compressa *(a << 1)* contiene prevalentemente le ***alte frequenze*** e dà un'immagine dettagliata su una piccola porzione del segnale *s(t)*



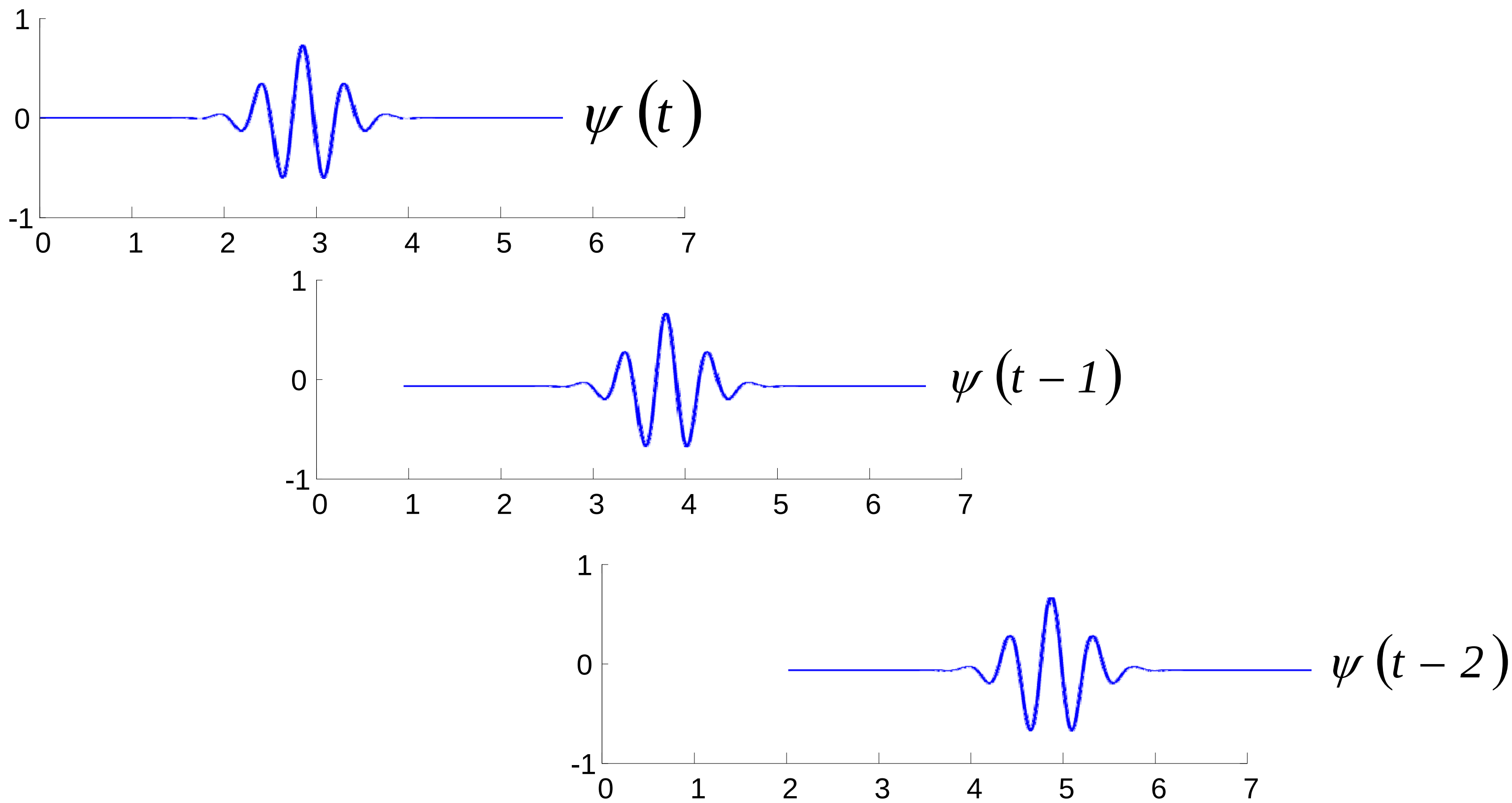
Scaling

- Lo *Scaling* di una wavelet consiste semplicemente nel suo stiramento o compressione
- Più piccolo è il fattore di scala, più compressa è la wavelet



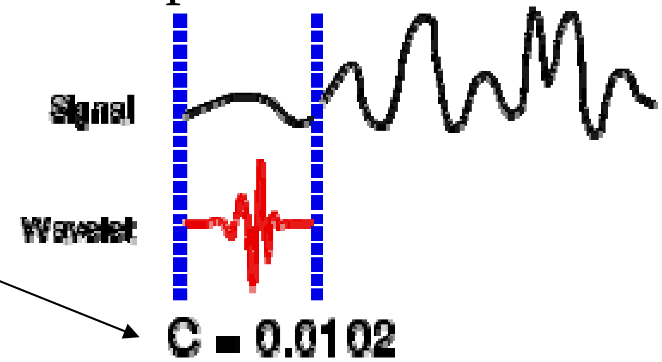
Shifting

- Lo *Shifting* di una wavelet consiste semplicemente ritardare (o anticipare) la sua posizione nel tempo
- Se $\psi(t)$ è la wavelet originale, $\psi(t-k)$ è la sua versione ritardata di k.

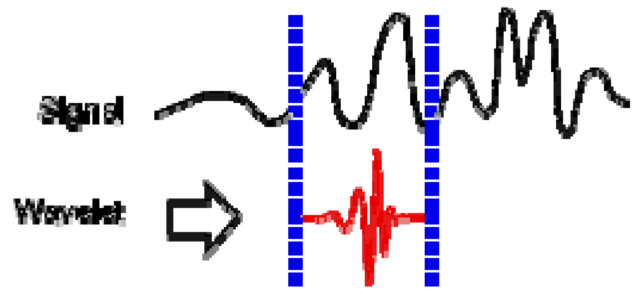


5 passi per costruire una CWT

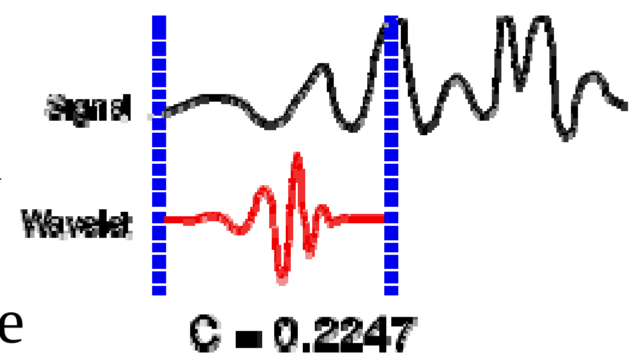
1. Scegliere una wavelet e paragonarla con una porzione del segnale
2. Calcolare il coefficiente *c* che rappresenta la *similarità* fra la porzione di segnale e la wavelet



3. Spostare la wavelet a sinistra e ripetere i passi 1 e 2 fino a coprire l'intero segnale



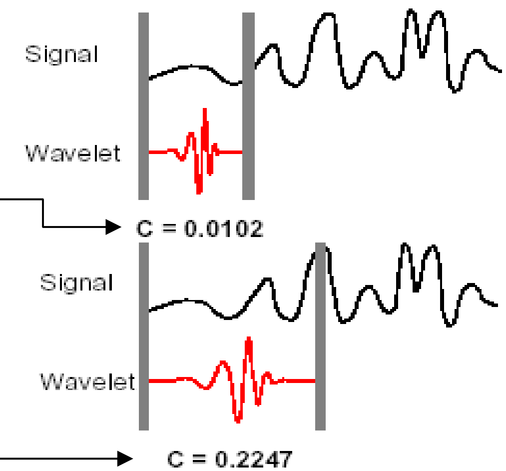
4. Scalare la wavelet e ripetere i passi da 1 a 3



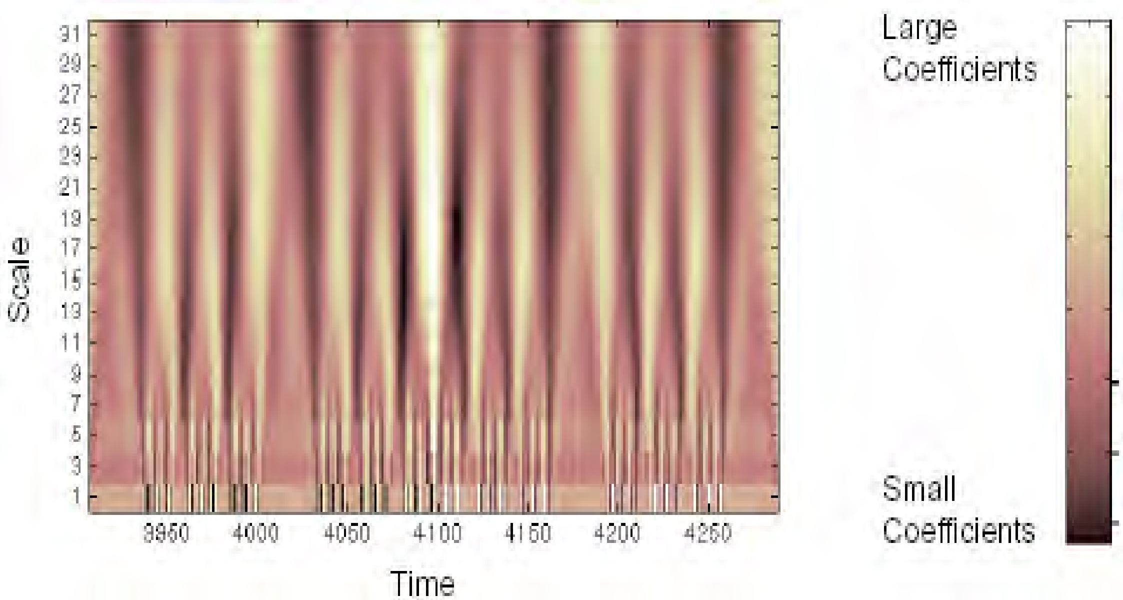
5. Ripetere i passi da 1 a 4 per tutte le scale

Significato della CWT

- La CWT è la somma di porzioni del segnale moltiplicate per versioni scalate e shiftate della wavelet.
- I *coefficienti* della wavelet sono il risultato di una *regressione* effettuata sul *segnale originale*.

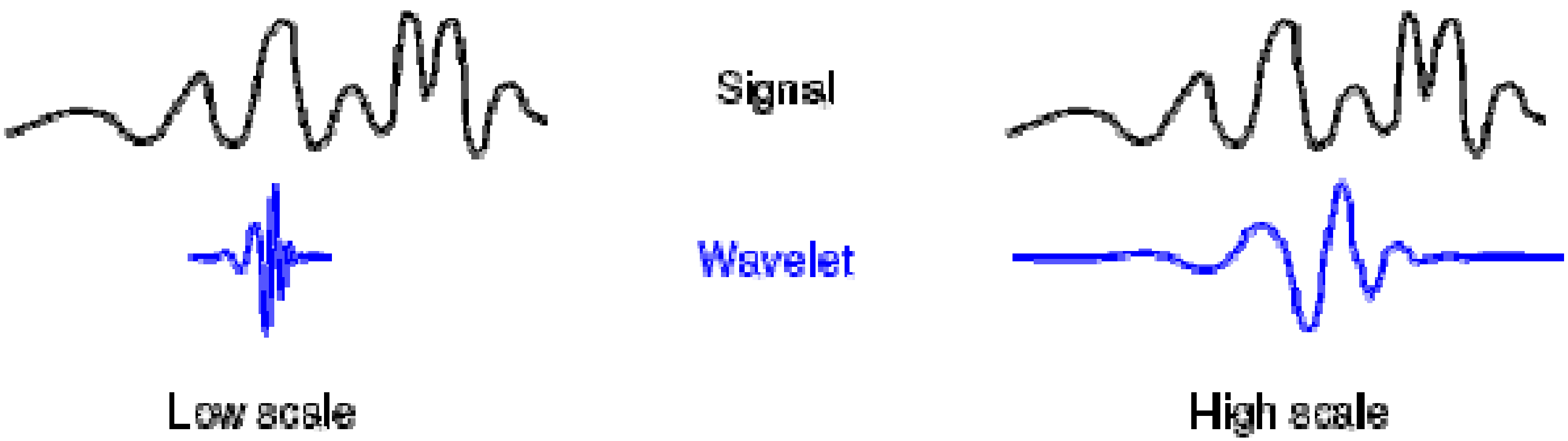


- Presentazione grafica dei coefficienti:
  - Si riporta in *ascisse* il *tempo delle osservazioni* e in *ordinate* la *scala* relativa a ciascu *coefficiente*;
  - Si associa un *pixel* di un *colore* associato alla posizione e al *modulo del coefficiente*



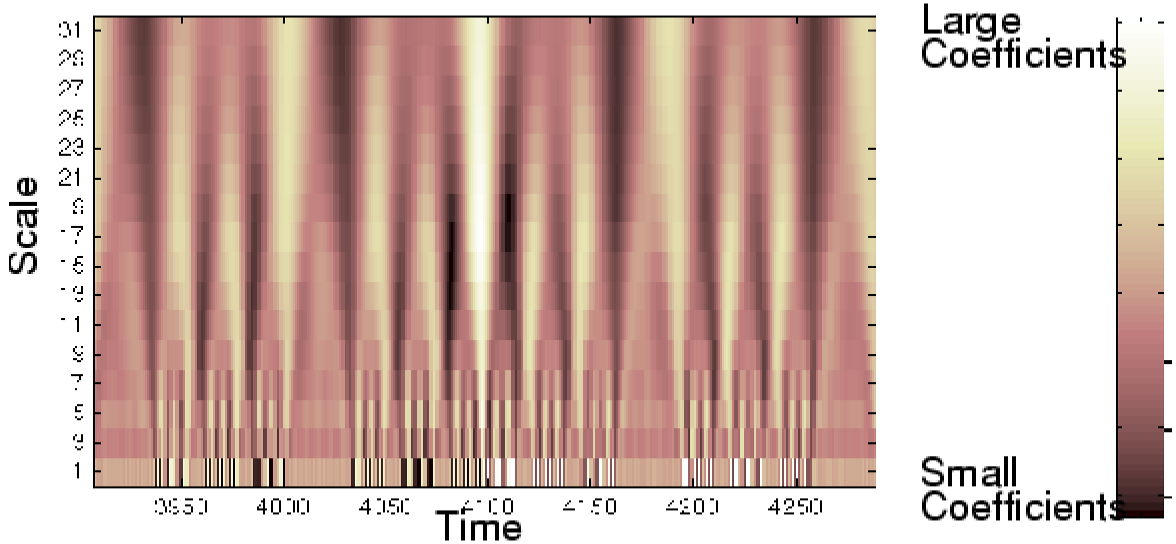
Scala e frequenza

- Le scale alte corrispondono alle wavelet più "stirate" (*stretched*).
- Maggiore è lo stiramento, maggiore è la porzione del segnale che viene comparata con la wavelet
- Perciò saranno più grossolani i dettagli del segnale che verranno rappresentati dalla wavelet



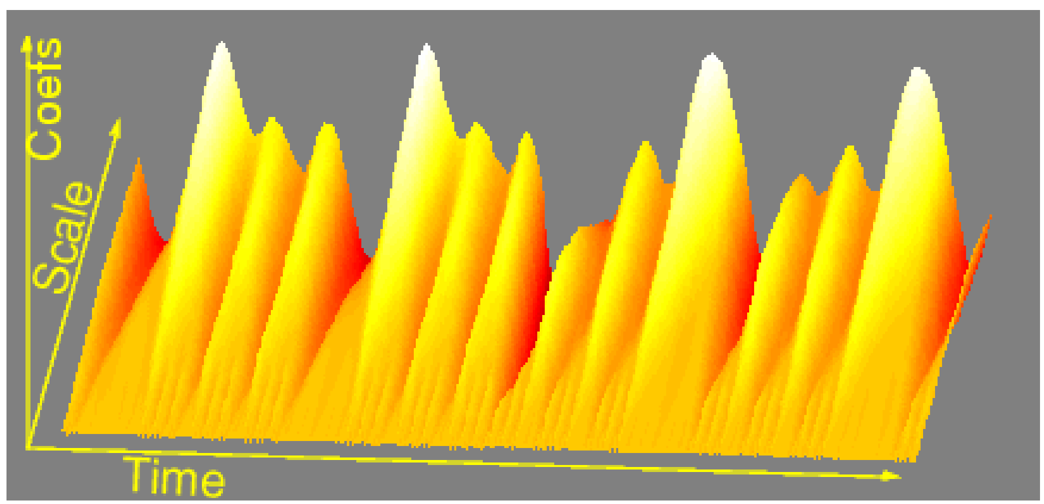
- Corrispondenza fra scala della wavelet e frequenza del segnale
  - Scala bassa →Wavelet compressa →Dettagli veloci →Alte frequenze
  - Scala alta →Wavelet dilatata →Dettagli lenti →Basse frequenze

Rappresentazioni dei coefficienti

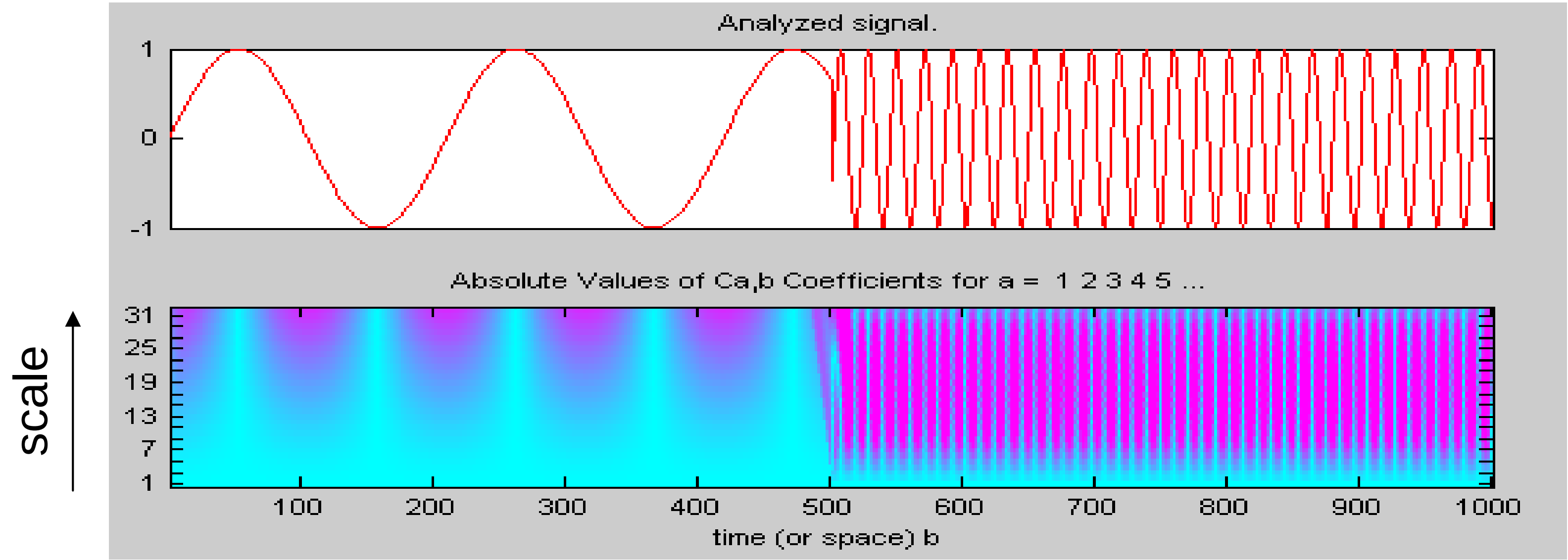


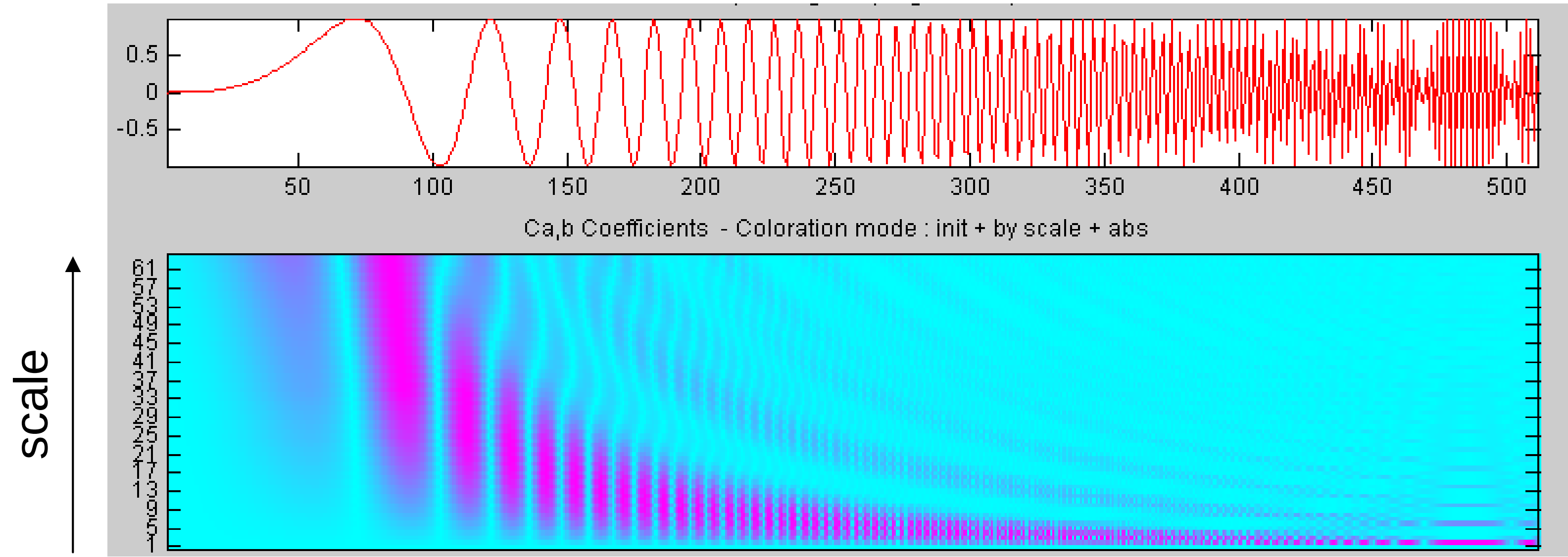
Rappresentazione
scala/tempo
bidimensionale

Rappresentazione
scala/tempo
tridimensionale



Esempi di WT Continua



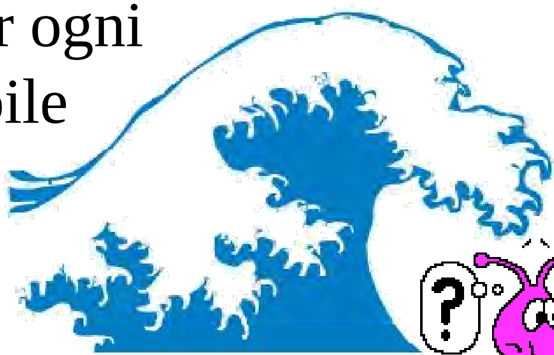


I coefficienti della trasformata wavelet continua (CWT) variano con il comporta-mento locale del segnale.

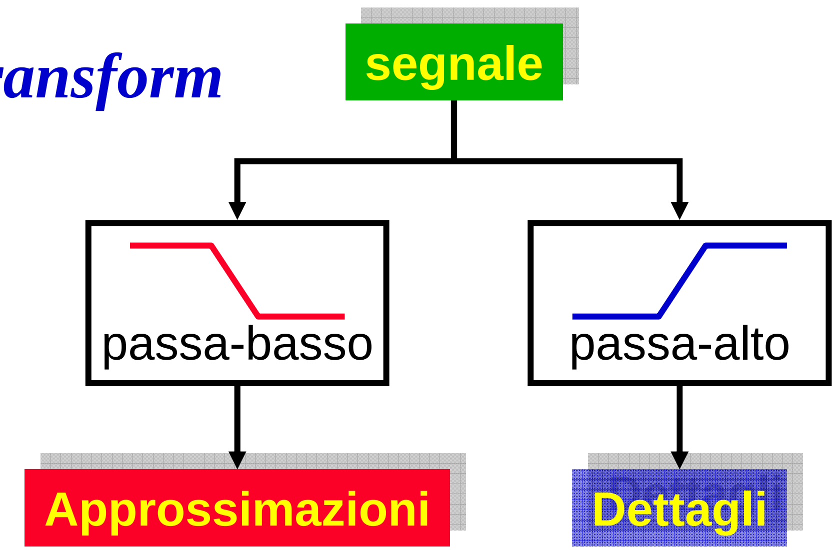
Spesso la quantità di informazione è ridondante.

La Discrete Wavelet Transform (DWT)

- Calcolare i coefficienti della CWT per ogni possibile scala è un compito formidabile e produce una quantità ridondante di dati.
  Meglio limitarsi ad un sottoinsieme

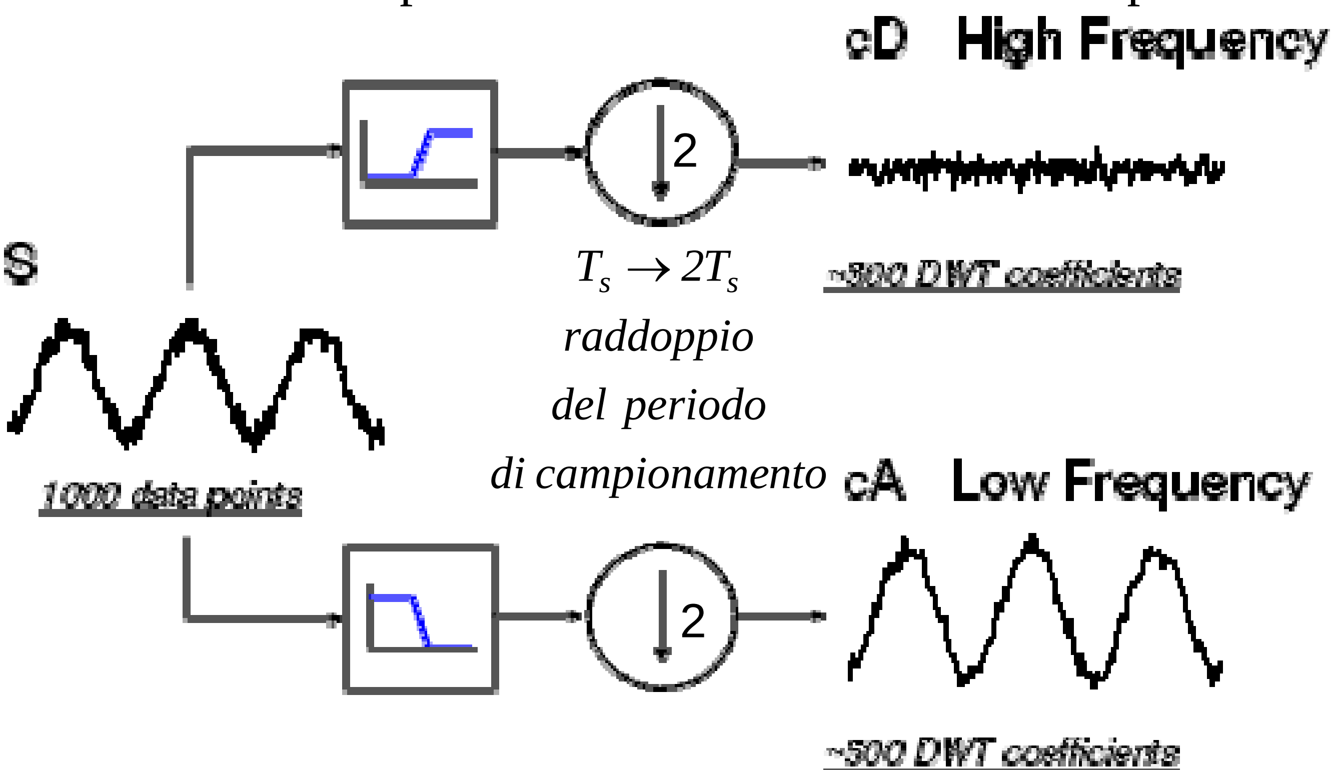


- Se si scelgono scale e posizioni basate sulle potenze di 2 (*scala diadica*), l'analisi è molto efficiente e non perde accuratezza
- Nasce così la ***Discrete Wavelet Transform***
- Essa decompone il segnale in
  - ***Approssimazioni:*** componenti di Bassa frequenza (scala alta)
  - ***Dettagli:*** componenti di Alta frequenza (scala bassa)



Il problema della proliferazione dei campioni

- Se dal segnale si ricavano per filtraggio due segnali (Approssimazioni e Dettagli) si raddoppia il numero di campioni
- Si ricorre al *downsampling*
  - Per ogni segnale filtrato si mantiene solo un campione ogni due
  - Si ha sempre lo stesso numero totale di campioni



Dettagli: contengono la parte ad ***alta frequenza*** del segnale

Approssimazioni: contengono la parte a basse frequenza del segnale

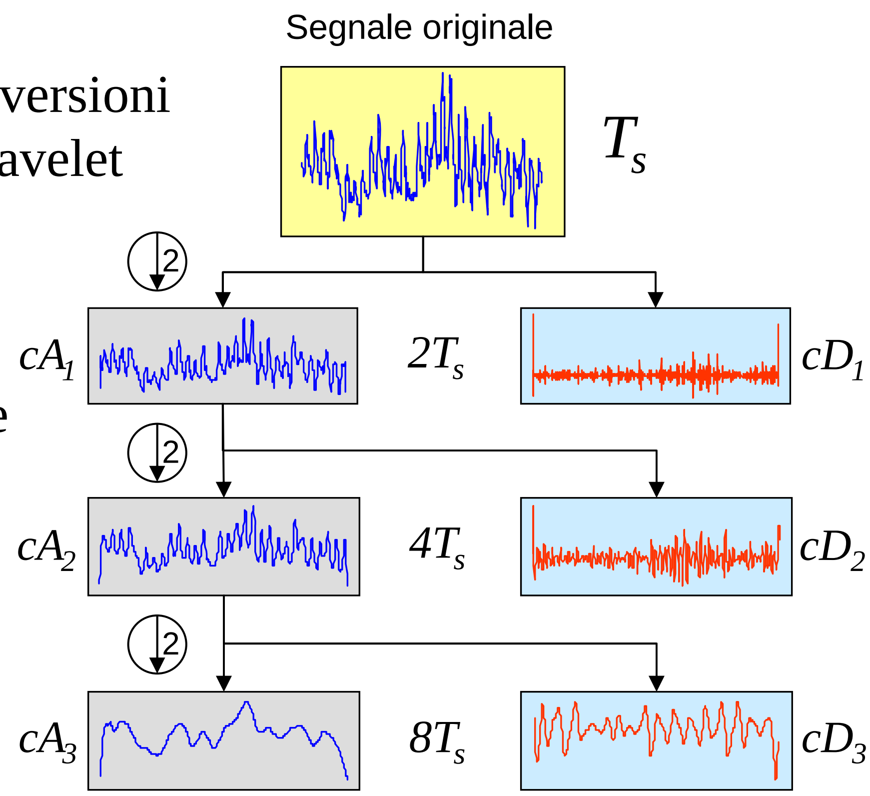
Trasformata wavelet discreta (DWT)

- Si limitano i fattori di scala a potenze di 2 (decomposizione diadica), ottenendo così la trasformazione wavelet discreta (DWT)

$$s(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{jk}\, 2^{j/2}\, \psi\left(2^{j} t - k\right)$$

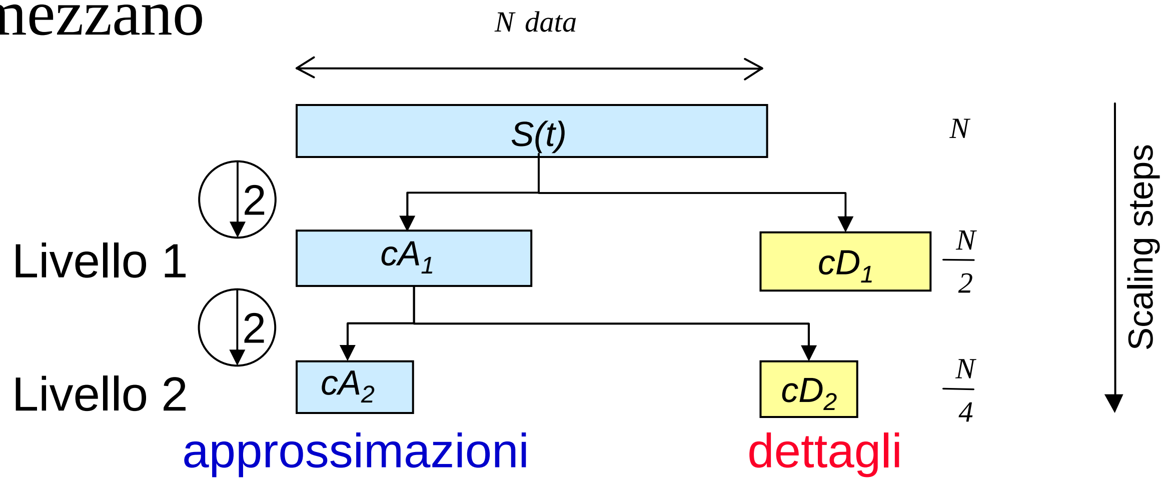
dove le funzioni wavelet $\psi(2jt - k)$ sono versioni scalate di $2j$ e traslate di k rispetto alla wavelet originale $\psi(t)$.

- Effettuando lo ***scaling diadico***, ad ogni decomposizione corrisponde un dimezzamento dei dati (downsampling) in base al campionamento T_s
- In tal modo si ha una decomposizione gerarchica del segnale in termini di Approssimazioni e *Dettagli*, però ad ogni livello successivo i campioni si dimezzano



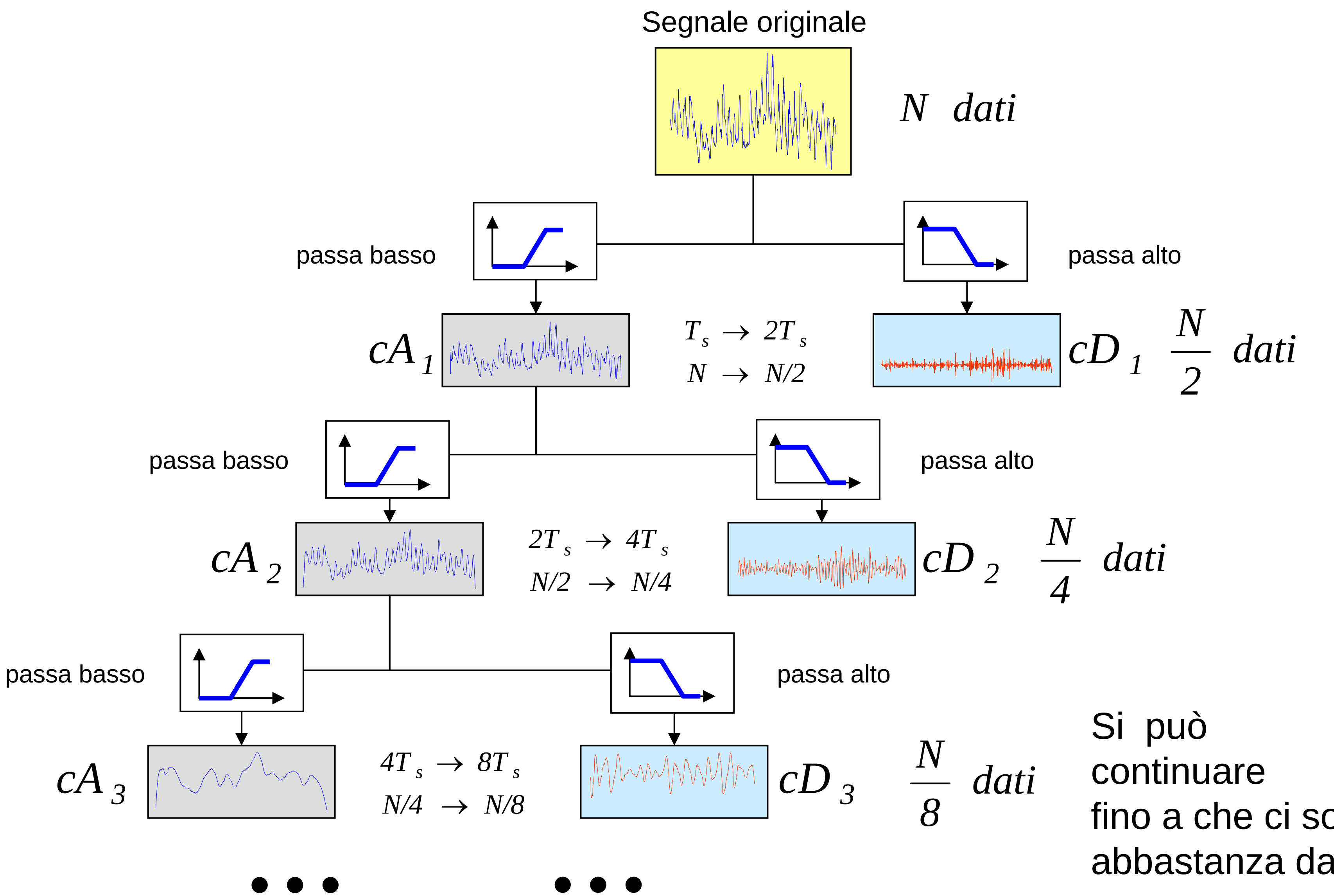
Considerazioni sulla DWT

- Procedendo per dimezzamenti successivi della scala si ottiene una ***Decomposizione Multilivello*** (MD) del segnale in successivi dettagli ed approssimazioni
- Ad ogni dimezzamento della finestra i dati disponibili si dimezzano



- Se la finestra conteneva N campioni, la decomposizione di primo livello ne conterrà N/2, quella di secondo livello N/4, quella di terzo N/8 e così via.

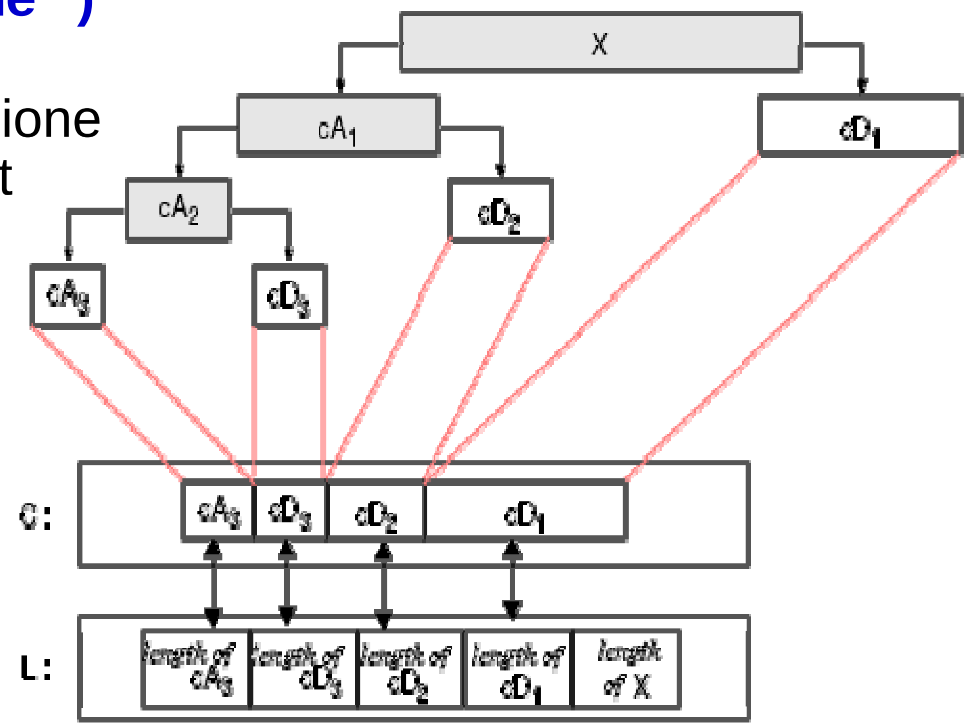
Decomposizione diadica della DWT



Si può continuare fino a che ci sono abbastanza dati

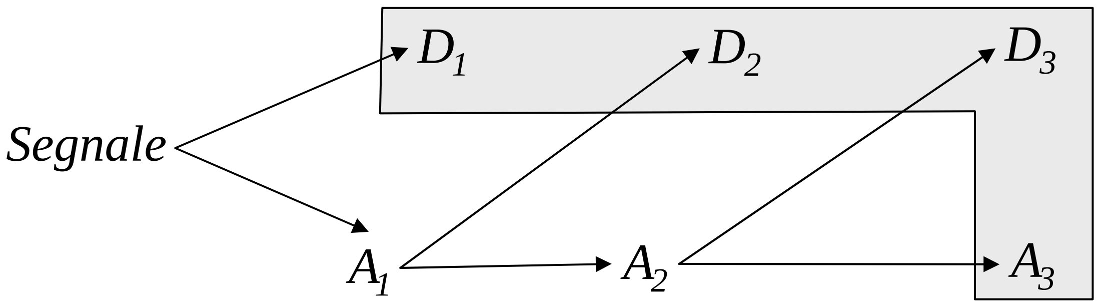
Wavelet Decomposition in pratica

- La decomposizione multilivello viene effettuata in Matlab con la funzione
  `[C,L] = wavedec(X,N,'wname‘)`
  - X = dati del segnale
  - N = Livello della decomposizione
  - wname = nome della wavelet
- Il comando restituisce l'insieme dei coefficienti delle approssimazioni e dei dettagli fino all'ordine N fissato e le dimensioni dei relativi vettori
  - C: insieme dei coefficienti
  - L: lunghezze dei vettori
  - I coefficienti si visualizzano con
    `appcoef(C,L,i); detcoef(C,L,i)`
  - **Le relative approssimazioni e dettagli con**
    `Ai=wrcoef(‘a',C,L,wname,i); Di=wrcoef('d',C,L,wname,i);`
  - *Nota: le approssimazioni e i dettagli così ottenuti hanno un numero di campioni uguale a quelli del segnale di partenza*



Algoritmo di decomposizione

- Convoluzione del segnale con filtro passa-basso (approssimazioni) e passa-alto (dettagli), seguito dal downsampling, ottenendo una struttura ad albero



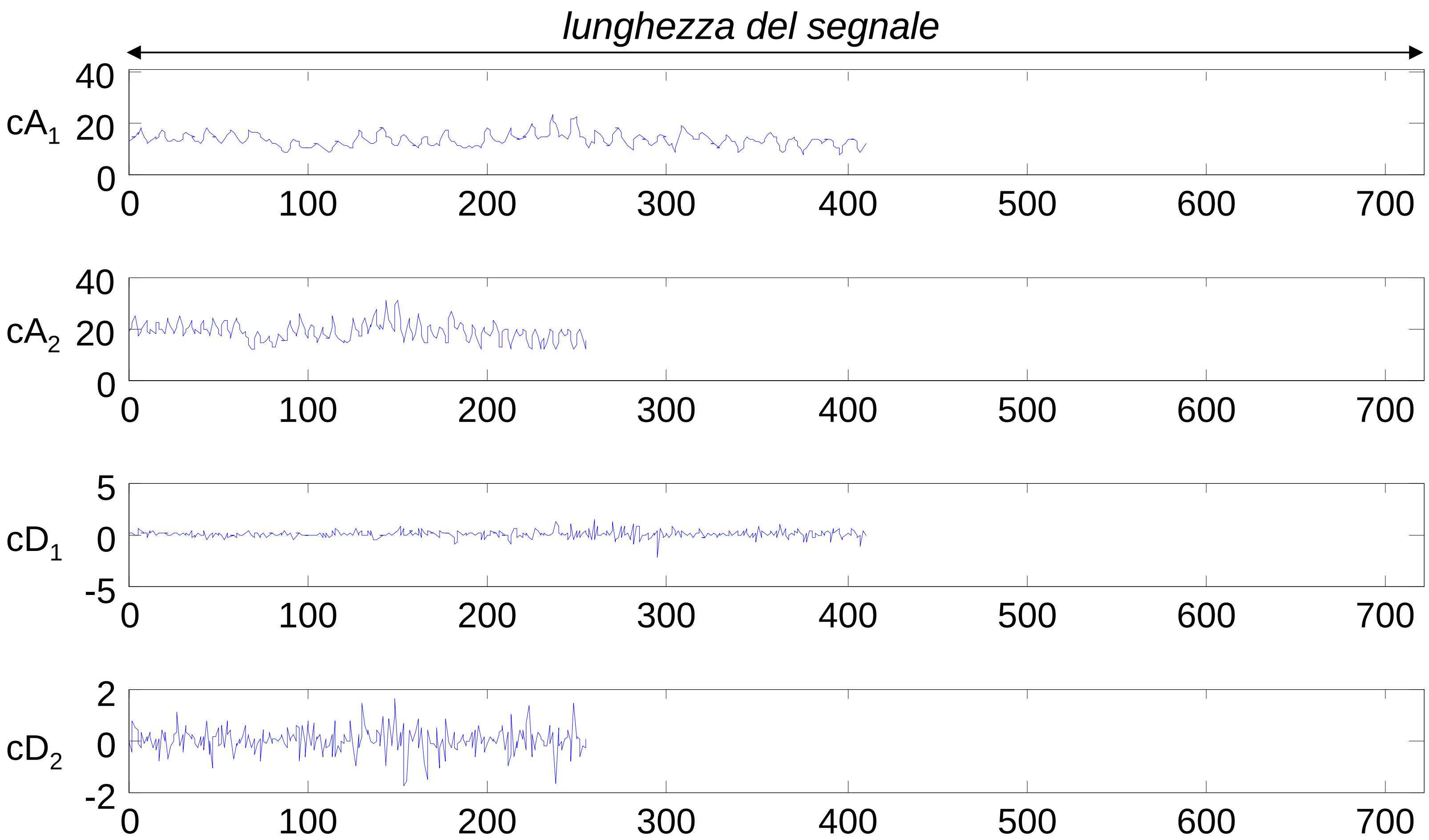
$$W_{dec} \sim \left(A_n, D_1, D_2, \ldots, D_n\right)$$

- La decomposizione wavelet W_{dec} è data dall'insieme dell'ultima approssimazione e da tutti i dettagli
- Se il segnale ha N campioni la decomposizione può avere al massimo $n = log_2 N$ livelli
- La funzione Matlab che effettua la decomposizione è

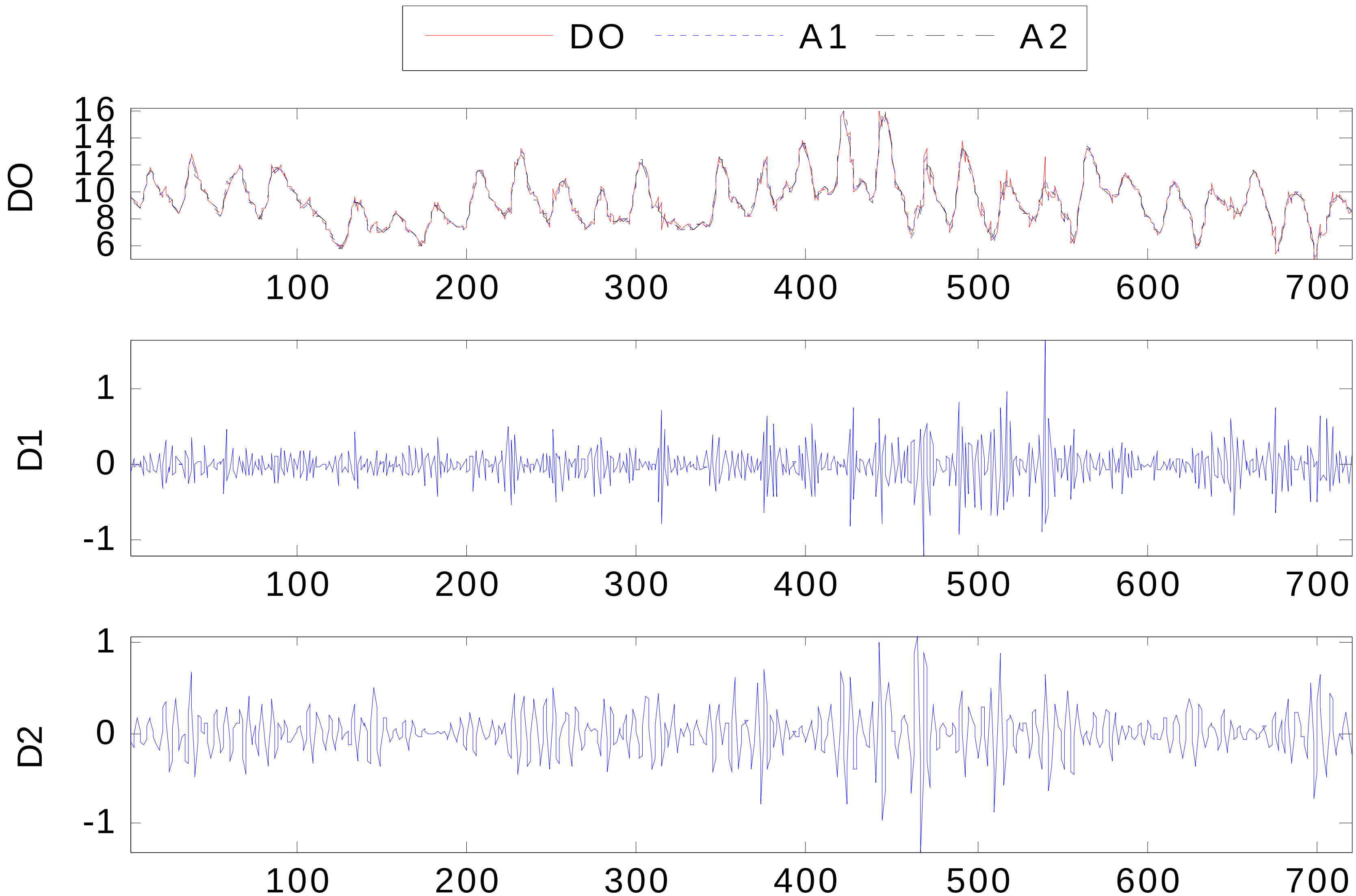
```
[C,L]=wavedec(DO,Level,wname);
```

Coefficienti della decomposizione a livello 2

Coefficienti della decomposizione wavelet del segnale di DO su 720 ore
Il numero dei coefficienti non è facilmente calcolabile e dipende dalla wavelet
Comunque a causa del downsampling decrescono al crescere del livello di decomposizione

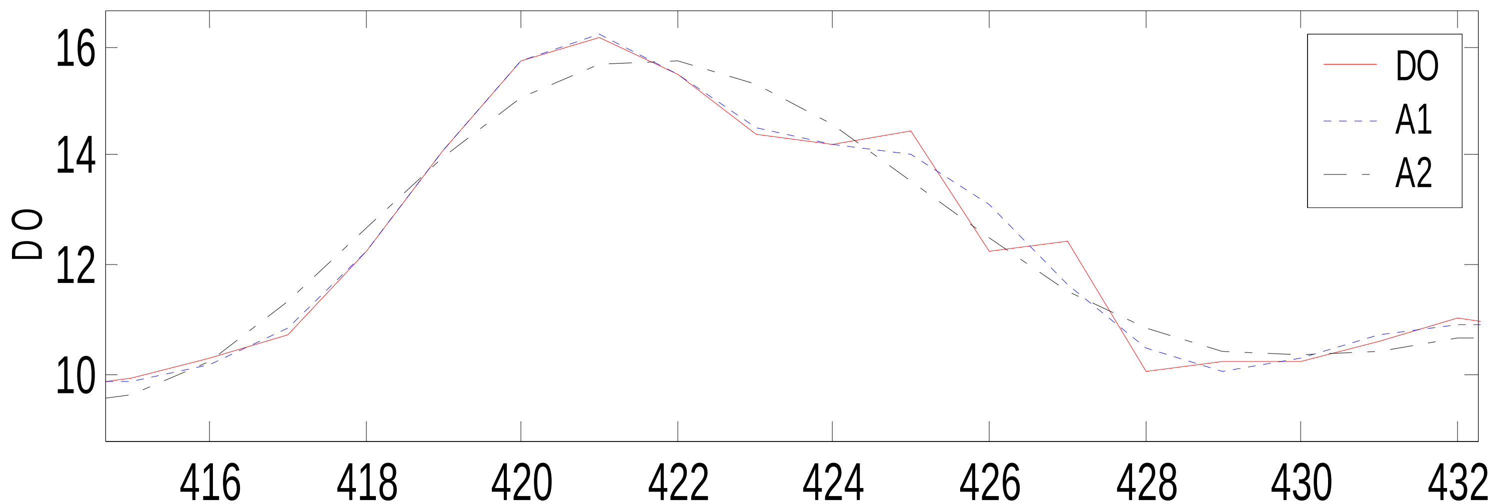


Decomposizione in approssimazioni e dettagli



Esame dettagliato delle approssimazioni

- Le approssimazioni effettuano uno smoothing del segnale proporzionale al grado dell'approssimazione
- Se il campionamento è di 1 h, l'approssimazione A_1 è basata sulle differenze a 2 h e la A_2 a 4 h
  - A_1 può rappresentare il segnale *"de-noised"*, supponendo che la fluttuazione casuale di 1 campione rappresenti l'effetto congiunto dell'errore della sonda e di variazioni ambientali di breve durata
  - A_2 rappresenta il segnale depurato dagli effetti ambientali di più lunga durata, mettendo in risalto il vero ciclo circadiano del segnale



Algoritmo di ricostruzione

- La ricostruzione del segnale avviene sommando l'ultima Approssimazione A_n con tutti i dettagli

$$S_{rec} = A_n + \sum_{i=1}^{n} D_i$$

- La relativa funzione Matlab è

```
S_rec=waverec(C,L,wname);
```

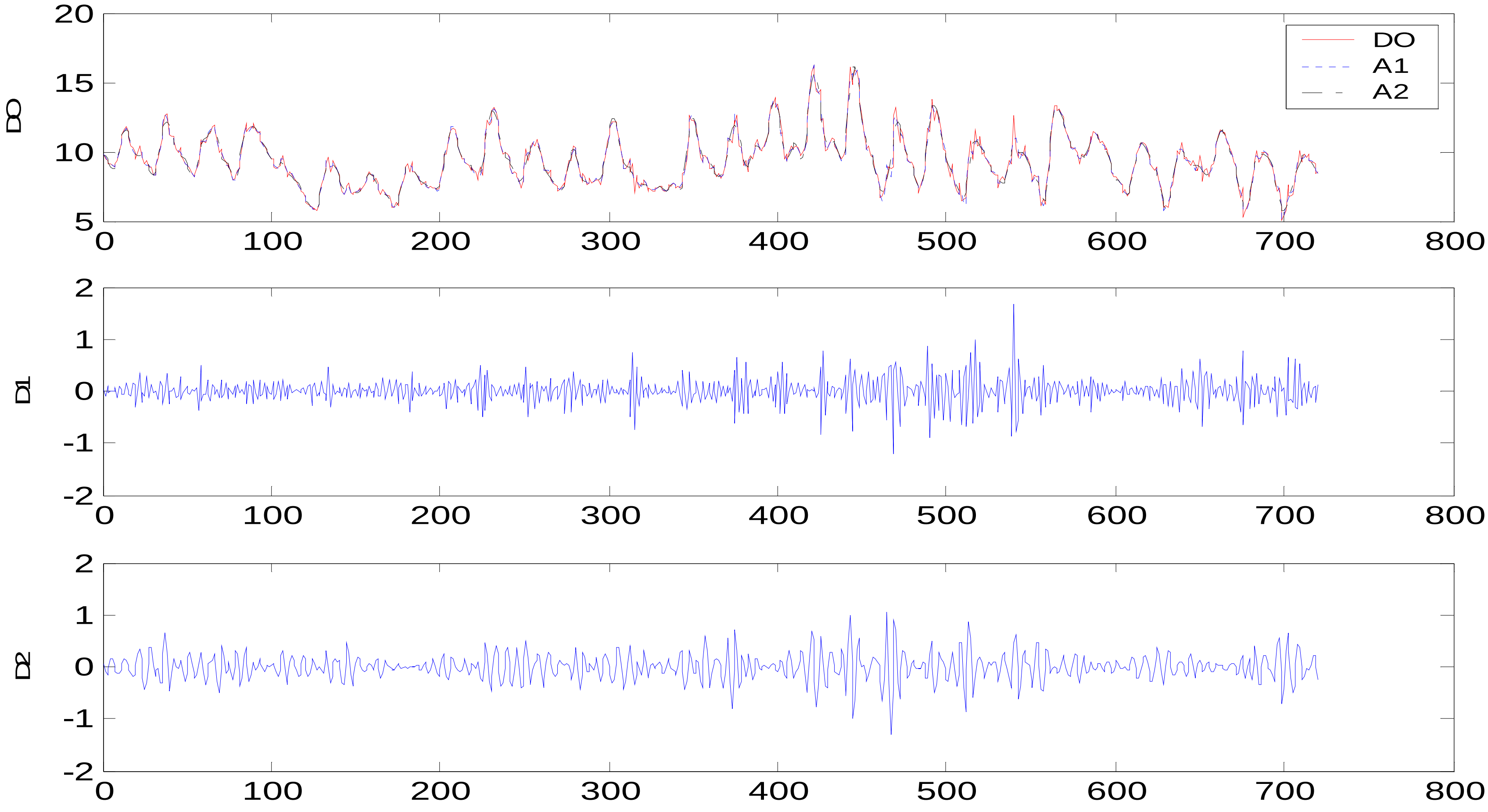
  - **C, L** sono i vettori della decomposizione (coefficienti e lunghezze)
  - **wname** è il nome simbolico della wavelet, che deve essere la stessa usata per la decomposizione
- Nota: qualsiasi è il numero dei coefficienti, approssimazioni e dettagli hanno gli stessi campioni del segnale di partenza
- Essi sono visualizzabili individualmente con la funzione

Ai=wrcoef('a',C,L,wname,i); i-esima approssimazione

Di=wrcoef('d',C,L,wname,i); i-esimo dettaglio

Esempio di decomposizione a livello 2

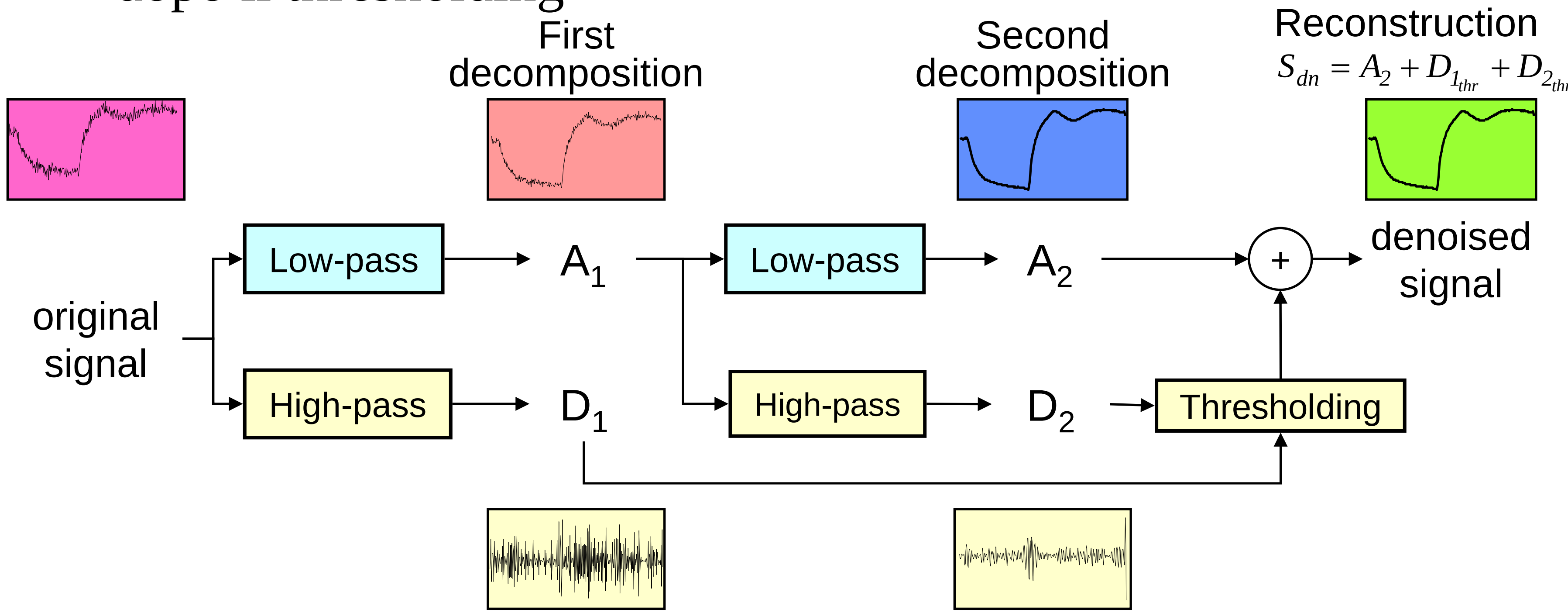
Segnale di ossigeno disciolto nella laguna di Orbetello, Aprile 2001. Tempo di campionamento T_s = 1 h



Denoising

- Si decompone il segnale fino al livello voluto (N)
- Si tagliano i coefficienti dei dettagli mediante operazione di soglia (thresholding) hard o soft
- Si ricostruisce il segnale sommando l'ultima approssimazione (A_N) con i dettagli dopo il thresholding

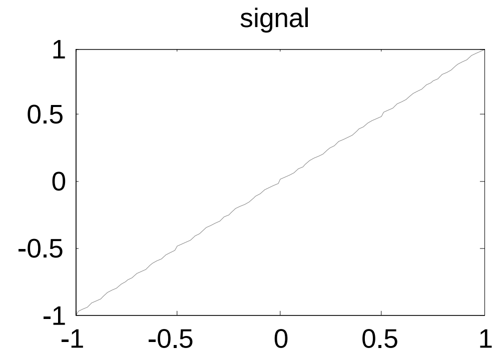
$$S_{dn} = A_N + \sum_{i=1}^{N} D_{i_{thr}}$$



Thresholding

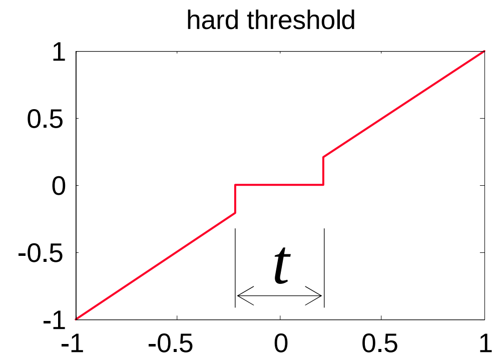
- Prima di ricostruire il segnale si trattano i coefficienti dei dettagli mediante thresholding

$$S_{dn} = A_N + \sum_{i=1}^{N} D_{i_{thr}}$$



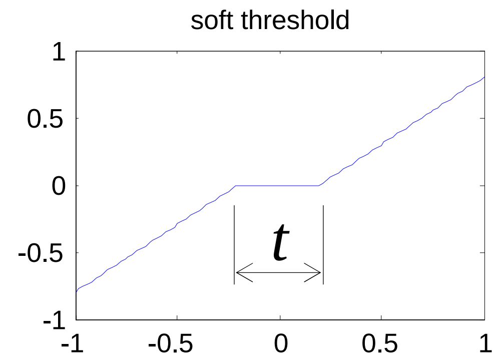
- ***Hard threshold:*** Pone a zero tutti i coefficienti in modulo minori della soglia

$$x_t = \begin{cases} x & se \ |x| > t \\ 0 & se \ |x| \le t \end{cases}$$

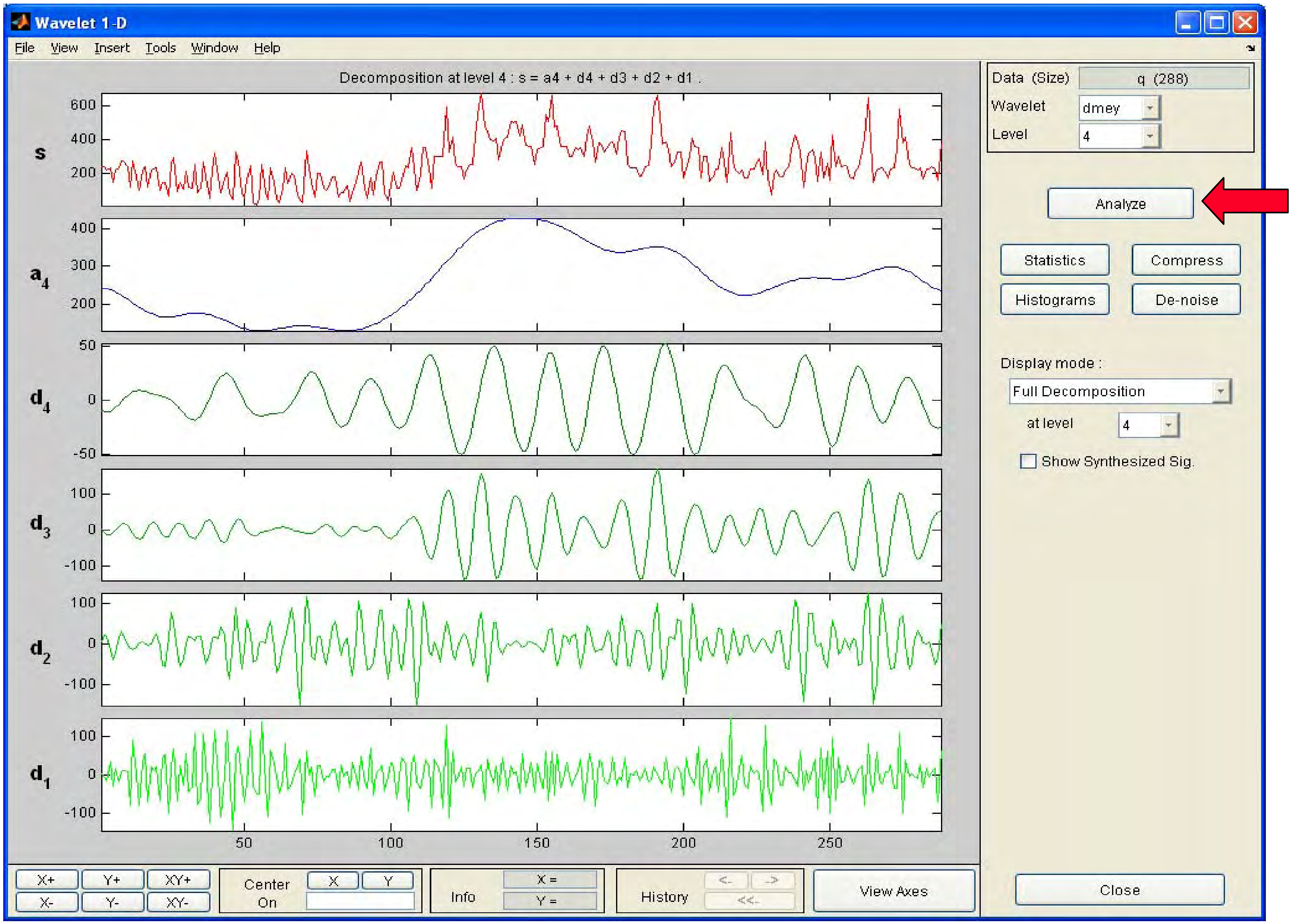


- ***Soft threshold:*** Dopo aver effettuato l'operazione di hard threshold, comprime verso lo zero i coefficienti non nulli

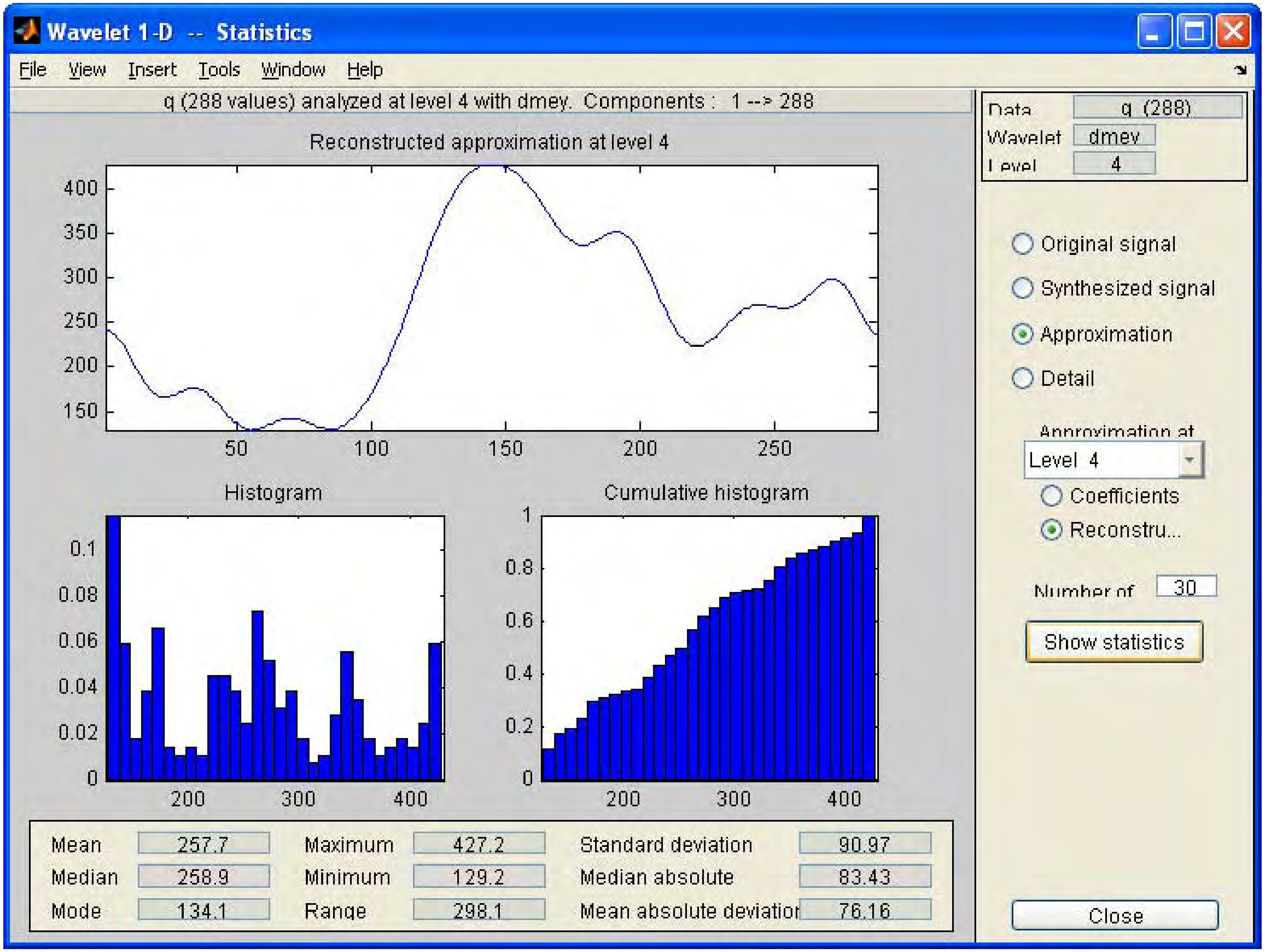
$$x_t = \begin{cases} x - t & se \ |x| > t \\ 0 & se \ |x| \le t \end{cases}$$



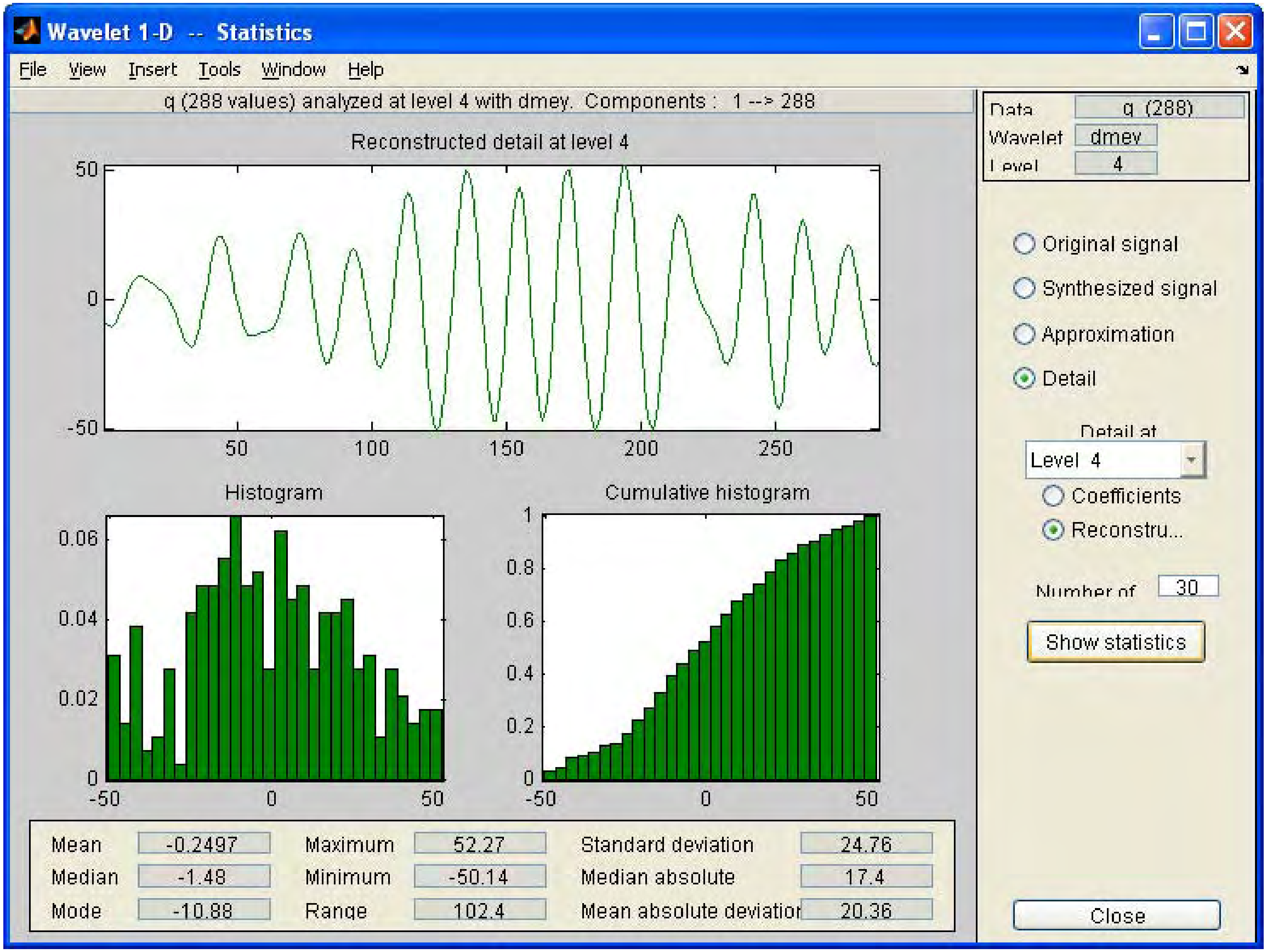
Wavemenu →Wavelet 1-D



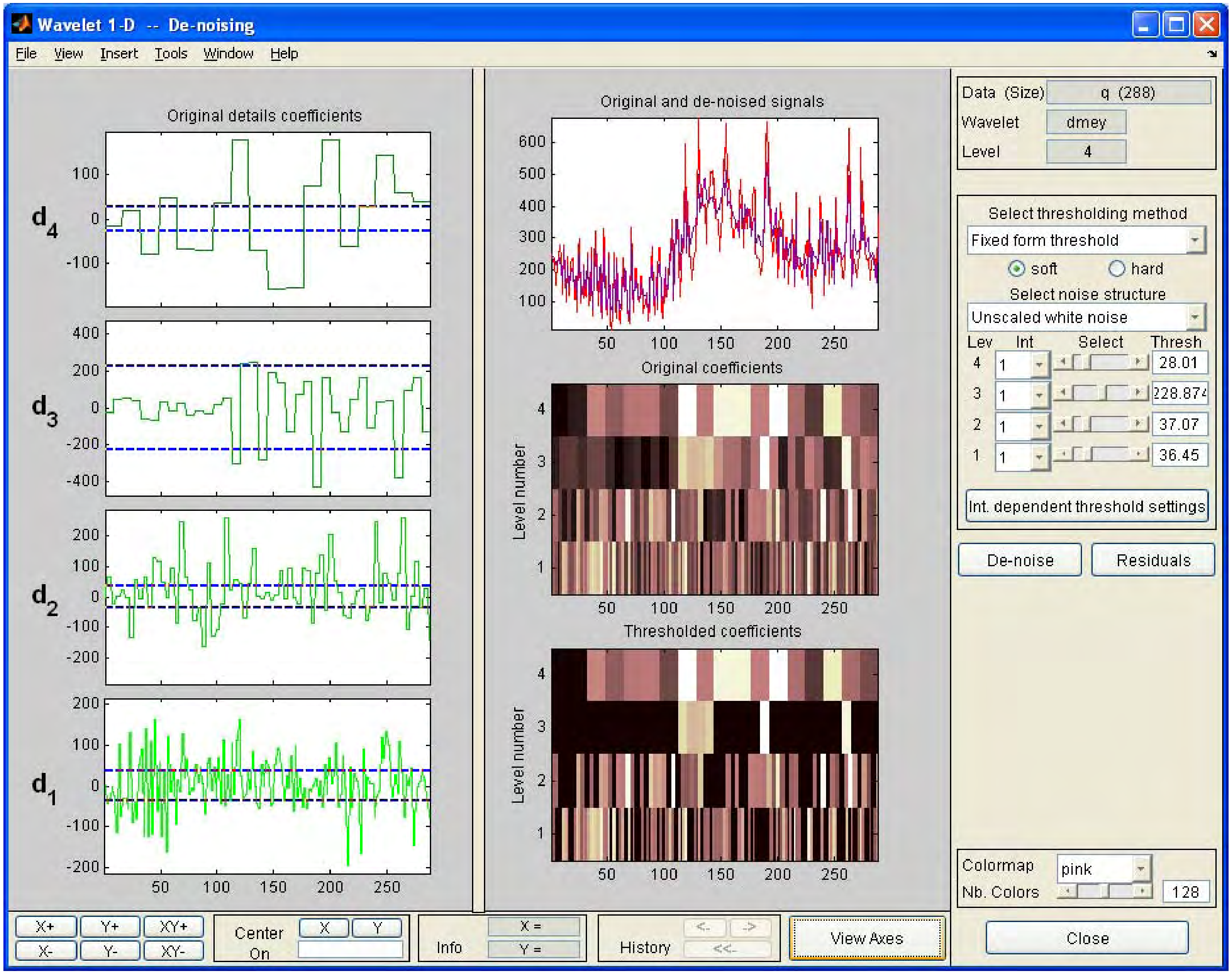
Wavemenu →Wavelet 1-D Statistics A4



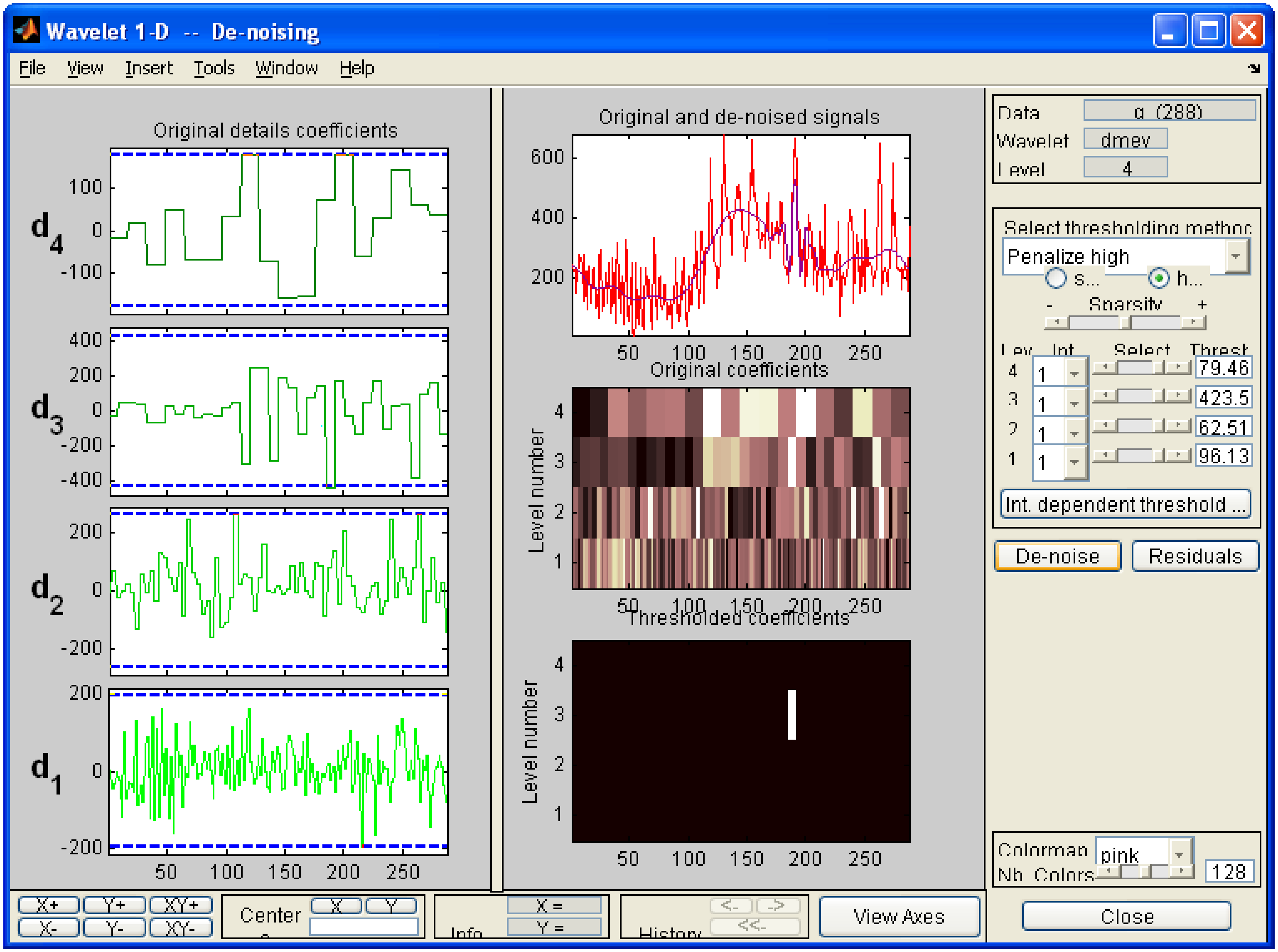
Wavemenu →Wavelet 1-D Statistics D4



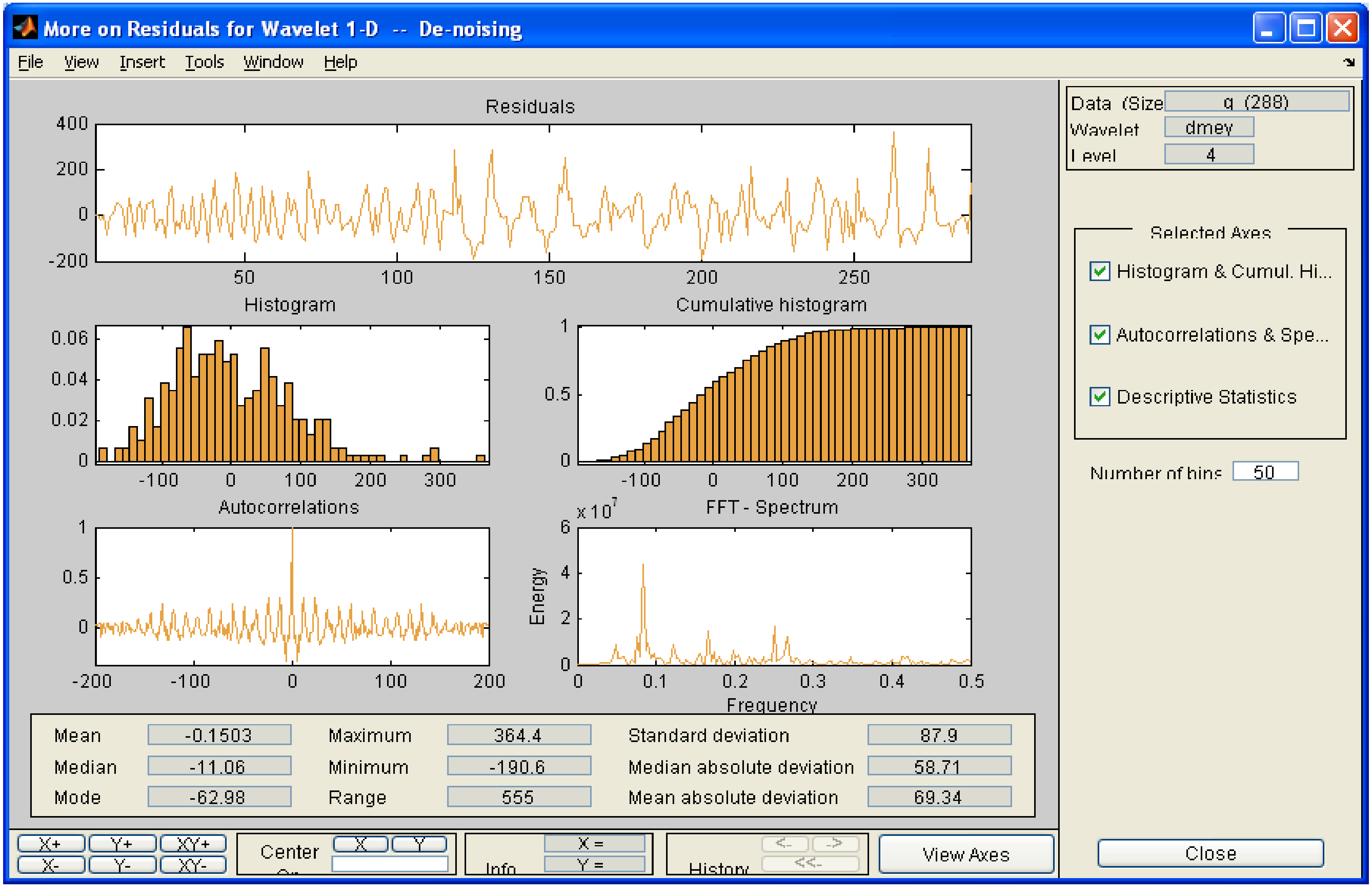
Denoising con soglie basse



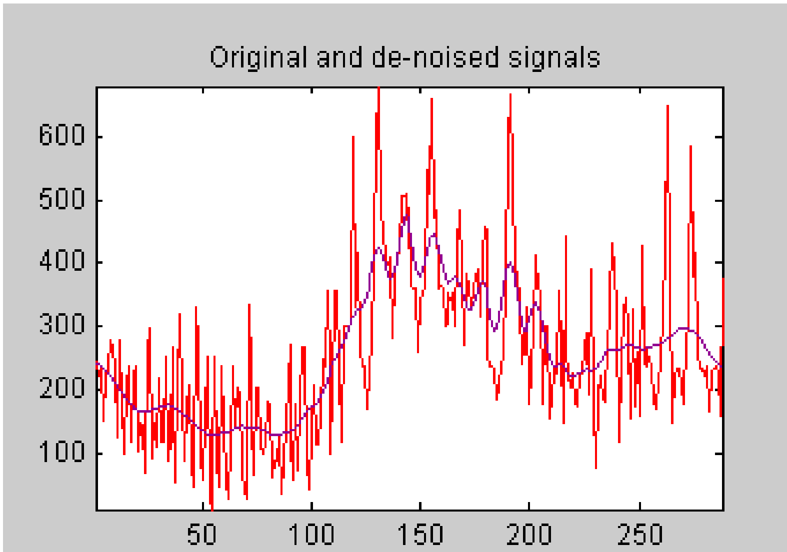
Denoising con soglie alte

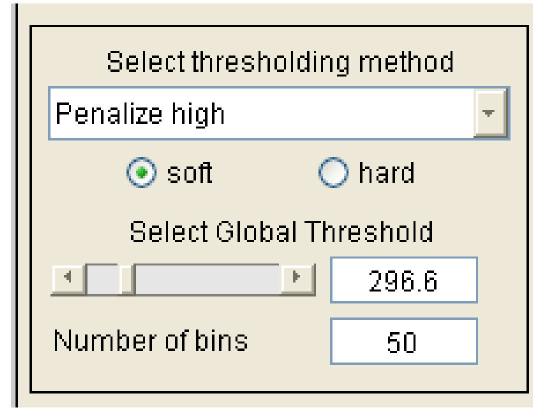


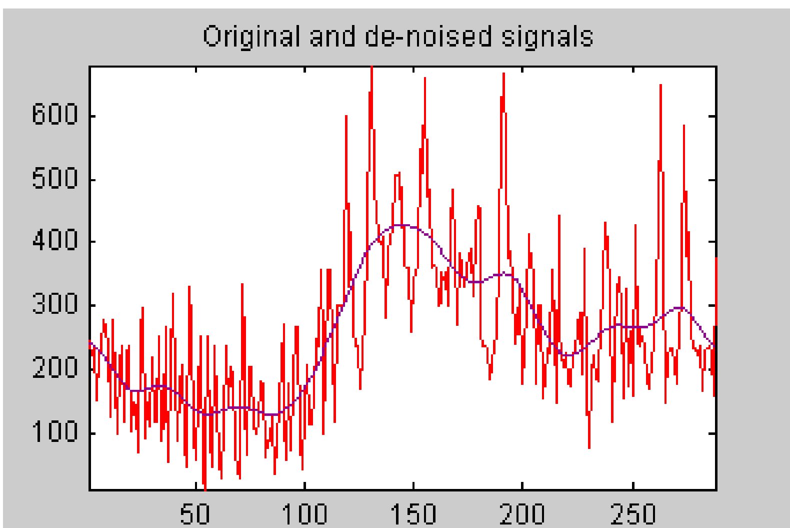
Denoising: Statistiche dei residui

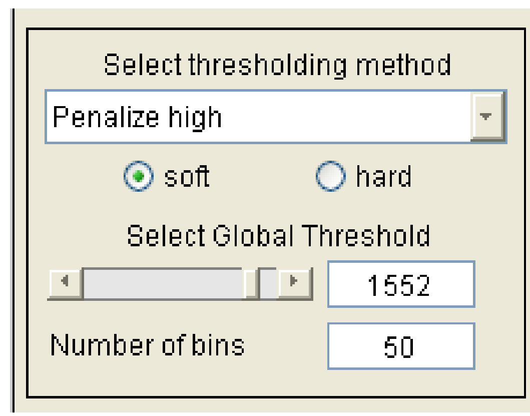


Effetto della soglia (thresholding)

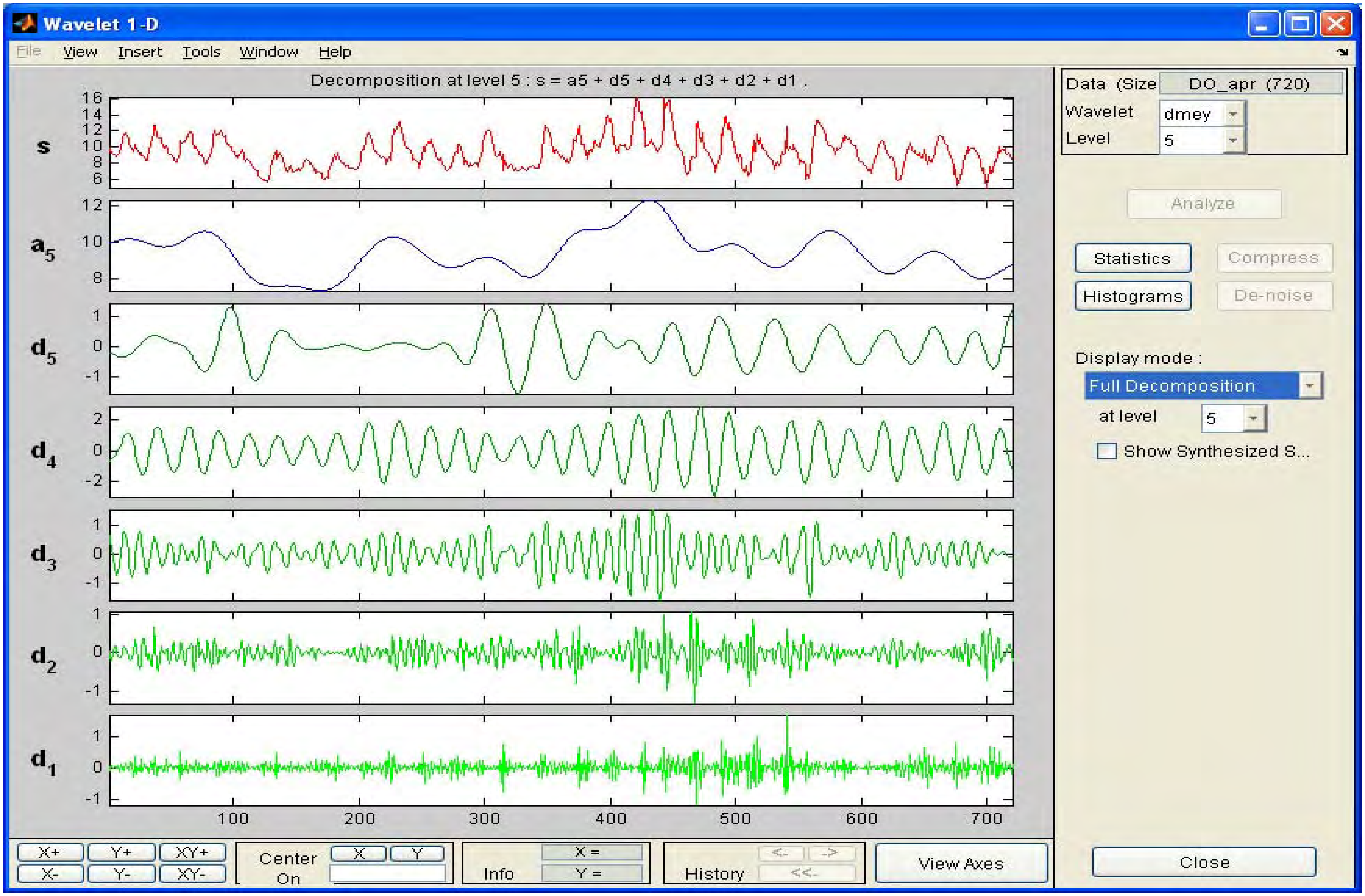




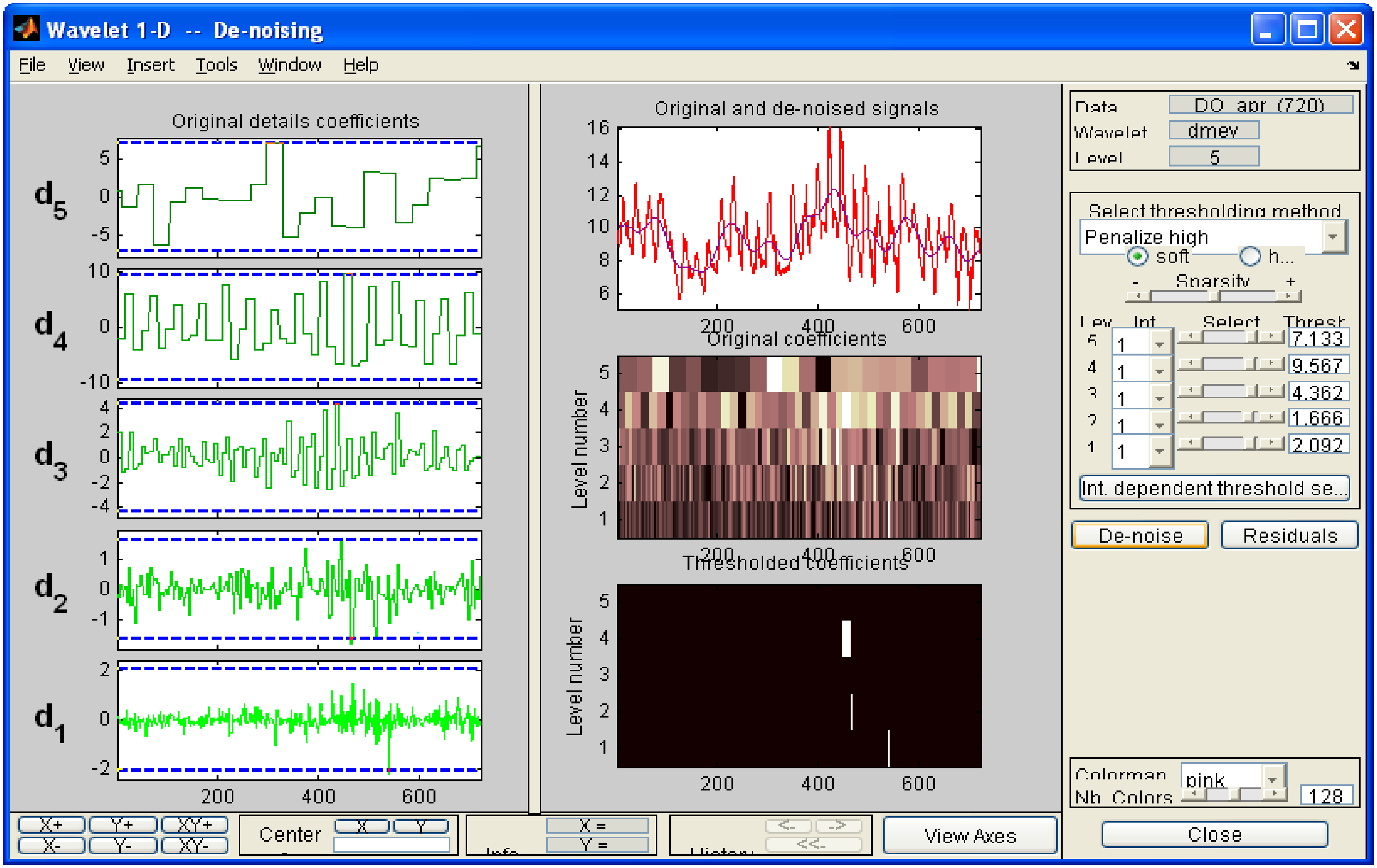




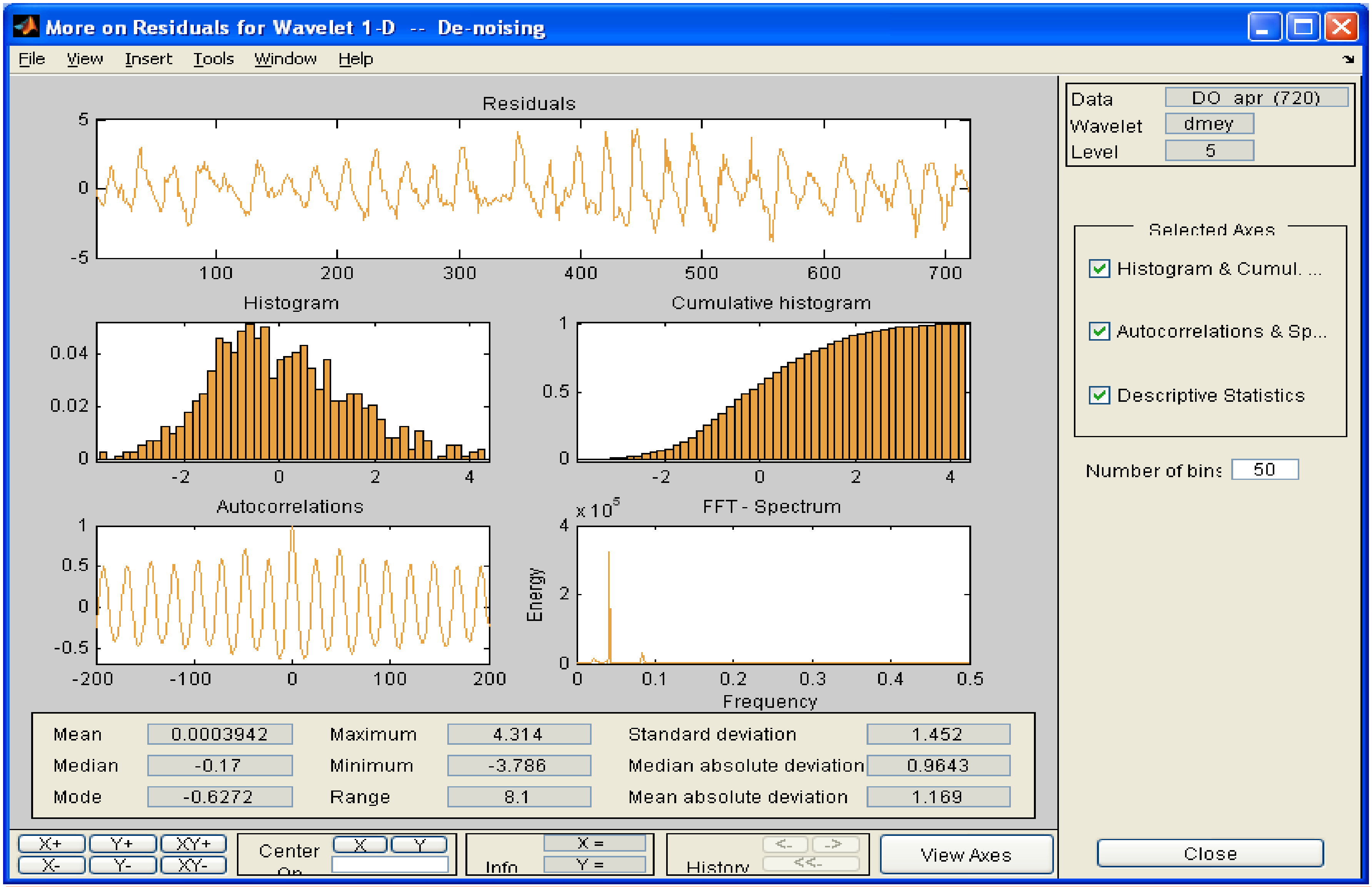
Serie di ossigeno disciolto



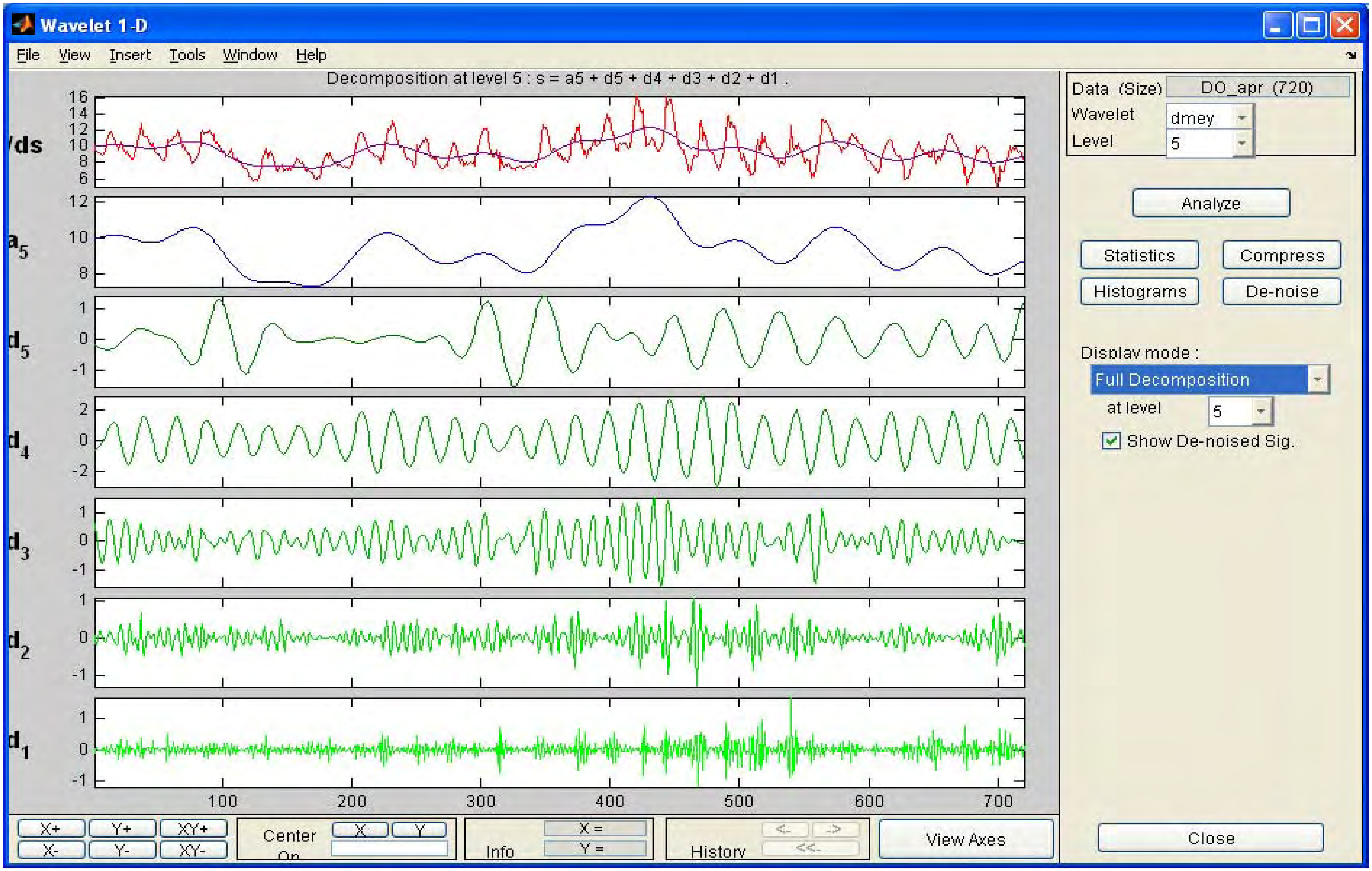
Denoising con soglie alte e penalizzazione alte freq.



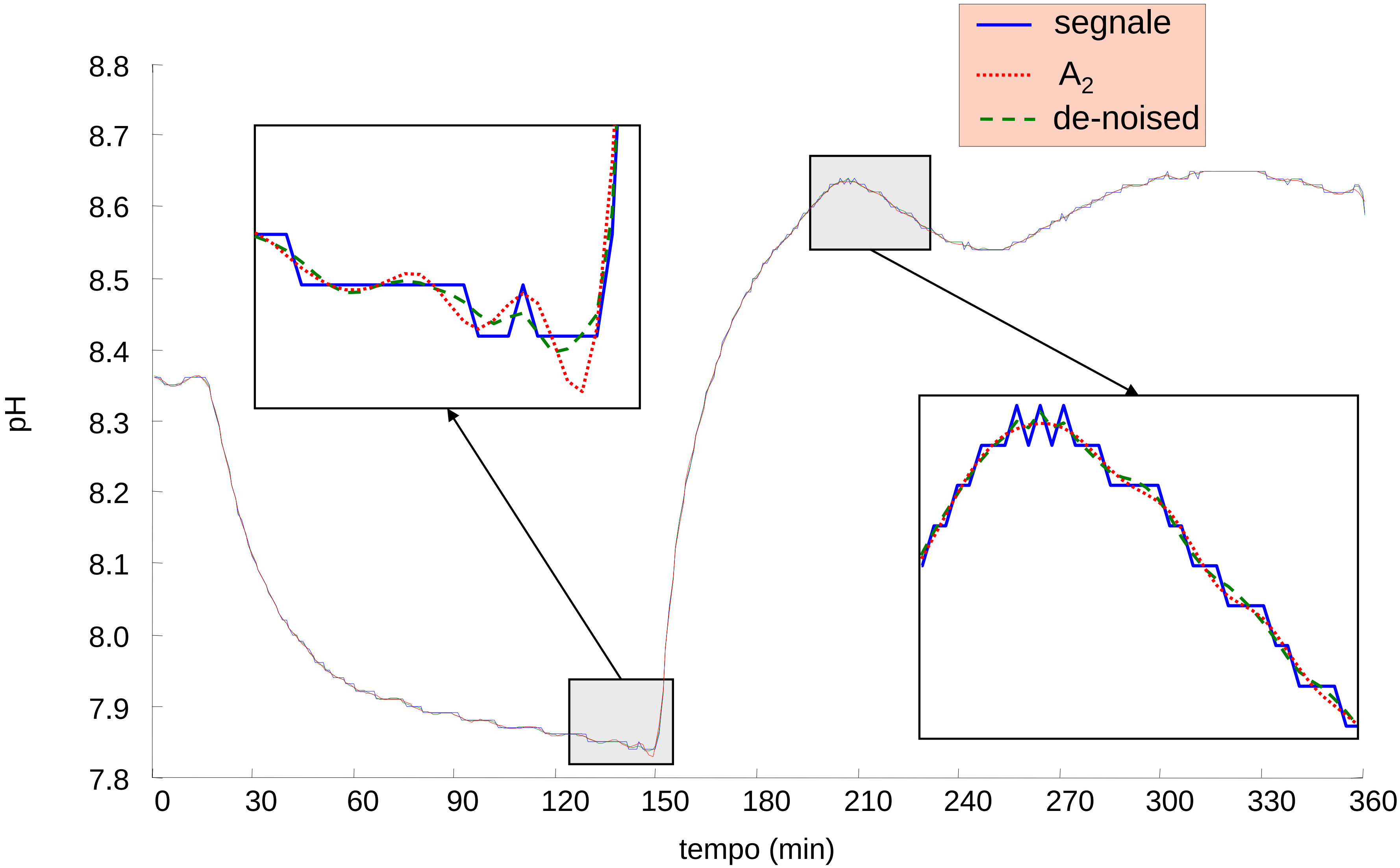
Statistiche dei residui



Risultato del denoising



Denoising di un segnale di pH



Varianti di denoising

- Partendo dalla decomposizione di base

```
[C,L]=wavedec(Dati,Level,wname);
```

- Si opera sui coefficienti dei dettagli applicando una soglia

```
C_den = wthcoef('t',C,L,N,[th(1) th(2)],'h');
```

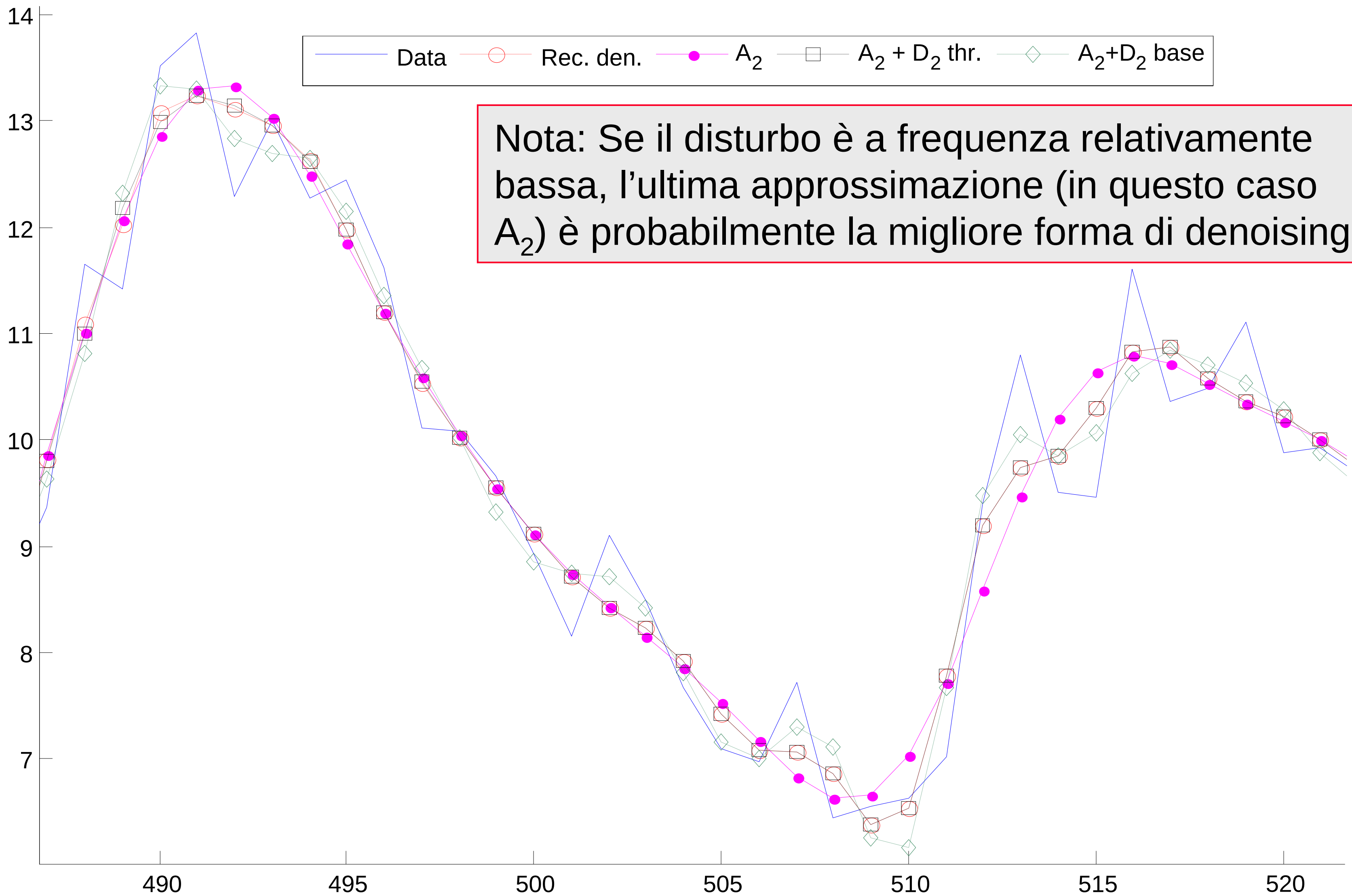
  - Il vettore **N** è l'indice dei dettagli $D_1,\dots D_N$
  - **th** è il vettore dei valori di soglia
  - **h** è il tipo di threshold (hard o soft)

- Si ricostruisce il segnale dalla wavelet con i coefficienti a soglia

```
Dati_den=waverec(NC,L,wname);
```

- Metodi di calcolo della soglia
  - Stein's Unbiased Risk (SURE) **rigrsure** $th = \sqrt{2\,ln\left(N\,log_2\,N\right)}$
  - Universal threshold **sqtwolog** $th = \sqrt{2Log\;N}$
  - Minimo del massimo errore quadratico medio **minimaxi**

Prestazioni dei vari denoising

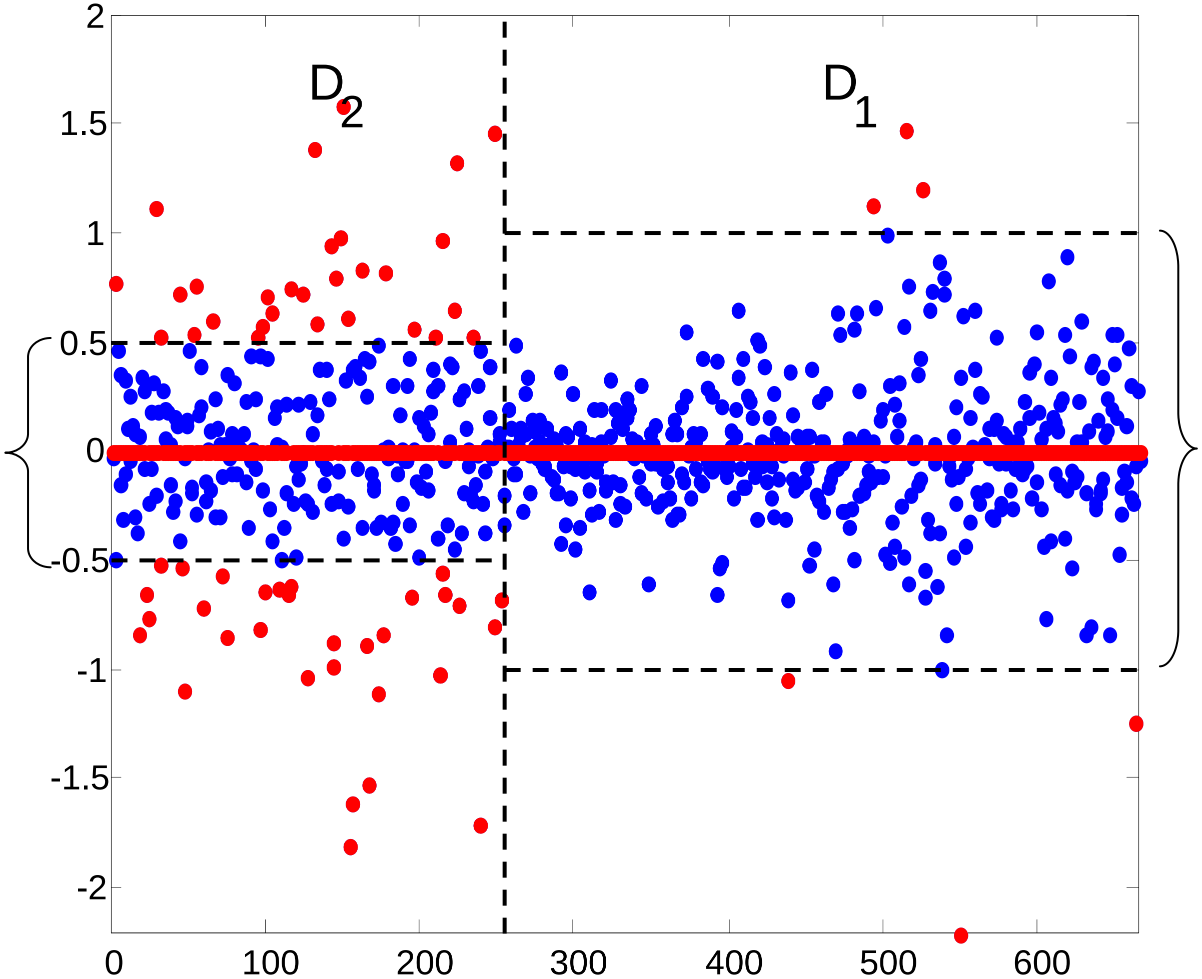


Nota: Se il disturbo è a frequenza relativamente bassa, l'ultima approssimazione (in questo caso A_2) è probabilmente la migliore forma di denoising

Effetto del thresholding ‘hard’ sui dettagli

● ● ● coeff. di dettaglio originali ● ● ● coeff. di dettaglio con soglia hard

I coefficienti di dettaglio D_2 entro questa banda sono ridotti a zero

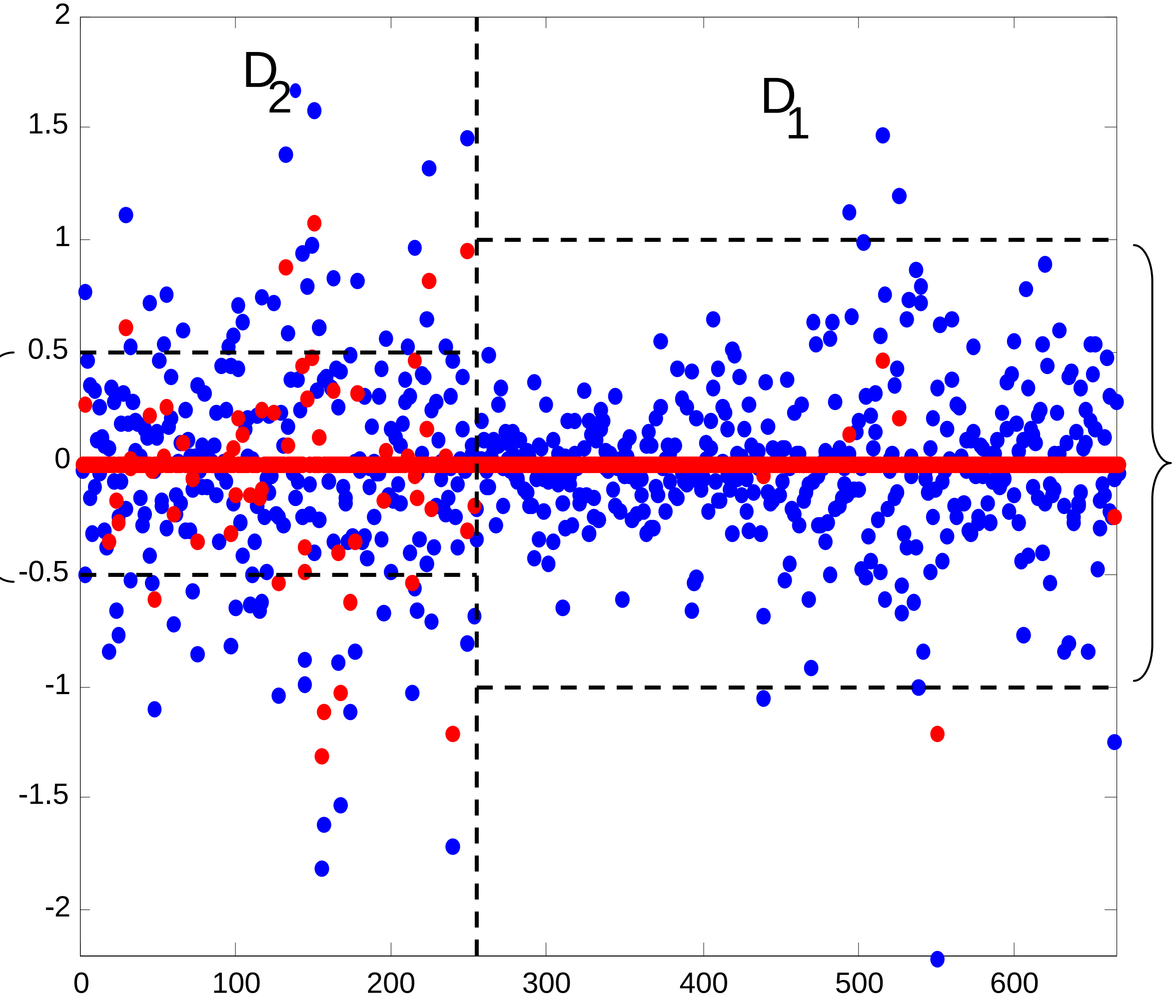


I coefficienti di dettaglio D_1 entro questa banda sono ridotti a zero

Effetto del thresholding ‘soft’ sui dettagli

● ● ● coeff. di dettaglio originali ● ● ● coeff. di dettaglio con soglia soft

I coefficienti di dettaglio D_1 entro questa banda sono ridotti a zero e quelli esterni sono compressi nella danda



I coefficienti di dettaglio D_1 entro questa banda sono ridotti a zero e quelli esterni sono compressi nella danda

Derivazione numerica di segnali

- Spesso l'informazione è contenuta nella derivata
- Derivare un segnale rumoroso è un suicidio
- Le proprietà di denoising delle wavelet producono segnali "derivabili"
- Preferibile alle Splines, anche se hanno anch'esse proprietà di smoothing, ma non controllano l'informazione contenuta nelle approssimazioni
- Le formule approssimate di derivazione numerica sono tanto migliori quanti più punti di appoggio usano

$$\frac{dx}{dt} \cong \frac{x(t+h)-x(t-h)}{2h} \qquad \frac{dx}{dt} \cong \frac{1}{12}\left[x(t-2h)-8x(t-h)+8x(t-h)-x(t+2h)\right]$$

$$\frac{d^2x}{dt^2} \cong \frac{x(t+h)-2x(t)+x(t-h)}{h^2}$$

$$\frac{d^2x}{dt^2} \cong \frac{1}{12}\frac{-x(t-2h)+16x(t-h)-30x(t)+16x(t+h)-x(t+2h)}{h^2}$$

- Per *h* ricordarsi che A_1 opera su dati a *2h* e A_2 a *4h*, perciò attenzione ad usare l'intervallo *h* giusto

Derivazione di segnali dopo denoising