

# Rapporti tecnici

# INGV

Analizzatore software per  
il formato *miniSEED*

# 285



## **Direttore Responsabile**

Stefano Gresta

## **Editorial Board**

Andrea Tertulliani - Editor in Chief (INGV - RM1)

Luigi Cucci (INGV - RM1)

Nicola Pagliuca (INGV - RM1)

Umberto Sciacca (INGV - RM1)

Alessandro Settimi (INGV - RM2)

Aldo Winkler (INGV - RM2)

Salvatore Stramondo (INGV - CNT)

Gaetano Zonno (INGV - MI)

Viviana Castelli (INGV - BO)

Marcello Vichi (INGV - BO)

Sara Barsotti (INGV - PI)

Mario Castellano (INGV - NA)

Mauro Di Vito (INGV - NA)

Raffaele Azzaro (INGV - CT)

Rosa Anna Corsaro (INGV - CT)

Mario Mattia (INGV - CT)

Marcello Liotta (Seconda Università di Napoli, INGV - PA)

## **Segreteria di Redazione**

Francesca Di Stefano - Referente

Tel. +39 06 51860068

Fax +39 06 36915617

Rossella Celi

Tel. +39 095 7165851

redazionecen@ingv.it

# Rapporti tecnici INGV

## **ANALIZZATORE SOFTWARE PER IL FORMATO *MINISEED***

Sandro Rao

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Centro Nazionale Terremoti)

# 285



## Indice

Introduzione.....	7
1. Cenni sul formato <i>miniSEED</i> .....	7
2. Il programma <i>miniSEED analyzer</i> .....	11
3. Considerazioni sul formato <i>miniSEED</i> .....	20
4. Sviluppi futuri.....	22
Bibliografia.....	22
Sitografia.....	22
Ringraziamenti .....	22



## Introduzione

Il formato *miniSeed* è un sottoinsieme del più articolato formato *Seed* (*Standard for the Exchange of Earthquake Data*), che nasce con lo scopo di standardizzare lo scambio di dati, relativi al campo sismologico, tra le varie comunità scientifiche. Gli attuali acquisitori sismici commerciali o costruiti nei laboratori elettronici di enti sismologici (come ad esempio la GAIA sviluppata interamente in ambito INGV), adottano il formato *miniSeed* per la trasmissione dei dati o la loro archiviazione in locale su diversi tipi di memorie di massa. Per tale motivo il *miniSeed* è divenuto presto oggetto di studio anche per progettisti elettronici o informatici al fine di comprenderne le principali funzionalità per poterle successivamente integrare nei sistemi hardware nella produzione del dato sismico o in quelli software per la sua interpretazione. A tale proposito sono stati sviluppati molti software per la visualizzazione di dati *miniSeed*, sia in ambiente *Unix* che *Windows*, utilissimi per il sismologo che dovrà effettuare un'analisi sismica, ma spesso con qualche carenza per lo sviluppatore che vuole indagare il dato prodotto nei minimi dettagli tecnico-informatici dal suo acquirente o dal suo software. Nel presente lavoro viene illustrato un piccolo programma (*miniSEED analyzer*), da me ideato e realizzato, in grado di analizzare dati *miniSeed* fornendone specifiche informazioni connesse con la struttura del formato stesso come ad esempio il livello di compressione del singolo "pacchetto" o la presenza di eventuali errori nello streaming preso in considerazione. Il software è in grado di aprire e di analizzare files *miniSeed*, di convertirli in formato *ascii* e di fornire una visualizzazione della forma d'onda o di estrarne determinate porzioni scelte in base ad una selezione temporale o di numero pacchetto.

Prima di entrare nello specifico del software, per comprendere al meglio le sue funzionalità, verrà illustrato brevemente il protocollo *miniSeed* in relazione al tipo di analisi che è possibile effettuare con il programma *miniSeed analyzer*. Seguiranno delle osservazioni sul livello di compressione in base ad alcune forme d'onda prese in considerazione.

Tengo a precisare che il presente lavoro non ha pretesa alcuna di sostituire altri softwares del settore (come il blasonatissimo *SeisGram2K* di Anthony Lomax, la *suite* di programmi del *Quanterra User Group* *qlib*, oppure i *tools* di *libmseed*), ma solo di fornire alcune funzioni di analisi che potrebbero esulare dal campo strettamente sismologico, ma fondamentali per eseguire dei test approfonditi di validità del dato prodotto da un acquirente.

### 1. Cenni sul formato *miniSEED*

Il formato *miniSeed*, come già detto, è un sottoinsieme semplificato del formato *SEED* nato negli anni ottanta del secolo scorso come soluzione al problema della standardizzazione dello scambio dei dati sismici e della successiva elaborazione, problematica resa più delicata dalla molteplicità dei sistemi operativi a corredo dei primi *computers*. Il *miniSeed* offre diverse modalità di rappresentazione del dato e diversi livelli di compressione dell'informazione a seconda dell'algoritmo impiegato, fra i quali il più adoperato è il cosiddetto *Steim1*. A questa modalità di rappresentazione (e di compressione) del dato si fa riferimento in questo rapporto tecnico.

L'organizzazione del formato e la relativa chiave di lettura è racchiusa all'interno di particolari "blocchetti" generalmente di 8 bytes dove vengono riportate informazioni sui dati sismici o su come questi devono essere letti ed interpretati. Nel *miniSeed* sono presenti solo due blocchetti denominati 1000 e 1001 che costituiscono l'ultima parte di un *header* fisso di 64 bytes, seguito dai campioni sismici costituenti il resto del "pacchetto" che è sempre dello stesso numero di bytes all'interno del medesimo streaming di dati. In figura 1 viene riportato per intero un pacchetto di 512 bytes, prodotto dalla stazione sismica SBPO della Rete Nazionale ed evidenziati con diversi colori i ruoli dei vari bytes, di seguito spiegati.

00000000h:	35 33 30 34 36 39 44 20 53 42 50 4F 20 20 20 48	: 530469D SBPO H
00000010h:	48 5A 49 56 07 DB 00 C6 00 00 00 00 06 40 00 CE	: HZIV.Ú.Æ.....@.Í
00000020h:	00 64 00 01 00 00 00 02 00 00 00 00 40 00 30	: .d.....@.0
00000030h:	03 E8 00 38 0A 01 09 00 03 E9 00 00 64 00 00 07	: .è 8.....é d
00000040h:	02 AA AA AA 00 3F 5C 8D 00 3F 61 4F FE C4 01 F5	: .aaa.? \   .?aOpA.ö
00000050h:	00 74 FF 49 02 F6 FF 4A FE AE 00 7B FE 39 FF 7F	: .tyI.öyJp@.{b9y
00000060h:	02 47 FE 45 FF CA 01 E1 00 15 00 9F 01 7B 00 99	: .GbEyÉ.á..... {.
00000070h:	FD 6D 00 6C FF 7F FE 06 00 DD FF 8C FF 08 01 2D	: ým.ly p..Ýý ý..-
00000080h:	2A AA AA AA 02 72 FE 84 01 9C 00 35 FC 33 01 68	: *aaa.rp ..5ú3.h
00000090h:	FD DE FE 78 01 AA 00 4E 00 4A 01 56 02 7D FE 85	: ýPpx.á.N.J.V.}p
000000a0h:	FF CA 01 1F FD 3B FE D0 00 F1 FF 4D 00 92 01 A4	: ýÉ. ý; pD.ñyM. .á
000000b0h:	FF 19 01 72 03 09 FD E1 FE 7C 01 3B FD 17 FF 6F	: ý. r. .ýáp  .: ý.ýo
000000c0h:	2A AA AA AA 00 F0 FD E9 01 86 01 7D 00 53 01 9E	: *aaa.šýé.   .} .S.
000000d0h:	00 80 FE 86 FE 92 FF A2 FF BA FE B5 FF B2 00 54	: . b b'ýcy²buý².T
000000e0h:	00 2C 01 EB FF A7 00 FD FF 81 FF AB FF C4 FE 25	: . .éyS.ýý ý«yÁp%
000000f0h:	01 0A FE 81 01 F8 FE DF 02 80 01 0D FD 8C 04 DA	: . .b  .øbB.   . ý  .Ú
00000100h:	2A AA AA AA FB 39 01 04 FE E2 FC C5 03 A3 FE A2	: *aaaú9. .páuA. fpc
00000110h:	01 FF FF 75 02 49 00 7B FD 84 02 38 FD C1 FE EF	: .ýyu. I. {ý  .8ýÁp
00000120h:	01 39 FE 7A FE E3 01 5C 01 B8 FF 03 00 BF 03 03	: .9bzbã. \ . ý. .é. .
00000130h:	FC 93 00 40 01 69 FC 5C FF B3 00 FE 00 D7 FF 0B	: ú  .@.iu\ý³. p.xý.
00000140h:	2A AA AA AA 03 B1 FF 88 FE 20 02 68 FC E7 00 43	: *aaa.tý p .huc.C
00000150h:	FF 79 FF E9 00 02 FE 57 01 4C 00 10 01 99 00 83	: ýývé. .pW.L. .   .
00000160h:	FF 53 00 C9 FE E4 FF DD 00 54 FE 23 01 47 FF BF	: ýS.ÉpáyÝ.Tp#.Gýé
00000170h:	01 23 02 0F FF 42 01 8D FE 6B 01 0F FE BC FE FA	: .#. ýB.  bk. .pkpú
00000180h:	2A AA AA AA 00 12 FD 31 03 7F FE F5 01 A1 01 8C	: *aaa. ý  . pö.   .
00000190h:	FF 02 04 20 FA ED FE 22 05 E9 FB B6 FC BD 05 34	: ý. . úip". éúfuk.4
000001a0h:	FF 0A FD E4 05 05 02 3F FB 10 00 D4 01 95 FA 6D	: ý.ýá. .?ú. Ó.  úm
000001b0h:	00 F9 00 DA FC B5 01 6A 03 6D FF F6 FF B1 01 60	: .ù.Úüp.j.myöyt. .
000001c0h:	2A AA AA AA FF DE FD 68 00 9E 01 0B FA E9 03 20	: *aaaýpöh.   . úé. .
000001d0h:	00 8F FC EB 04 82 00 0B 00 BB 00 98 FF E4 FD D6	: .  ué.   . . . .  yáyÖ
000001e0h:	FE 90 01 F2 FD 01 01 FD 02 2A FF 23 02 71 00 99	: p  .óý. ý. *ý#.q.
000001f0h:	00 21 FF A8 01 2D FF 7D FD FC 00 A8 FD E7 00 E3	: .  ý. .-ý}ýi. .ýc.ã

**Figura 1.** Pacchetto dati *miniSeed* numero 530469 proveniente dalla stazione della Rete Sismica Nazionale Ingv (IV) SBPO, canale HHZ a 100 campioni al secondo. Il primo dato associato all'orario 00:00:00 del 17/07/2011 si ha dopo 16 campioni, come spiegato nel testo.

Nella successiva tabella 1 vengono analizzati i vari campi dell'*header* con riferimenti di esempio presi dal pacchetto riportato in figura 1.

Posizione byte	Esempio	Nome campo	Lunghezza (bytes)
0-5	530469	Numero pacchetto	6
6	D	Indicatore tipo dati	1
7		Riservato	1
8-12	SBPO	Codice stazione	5
13-14		Locazione stazione	2
15-17	HHZ	Codice canale	3
18-19	IV	Codice rete	2
20-29	Vedi tab. 2	Orario pacchetto	10
30-31	00 CE = 206	Numero campioni pacchetto	2
32-33	00 64 = 100	Campionamento	2
34-35		Moltiplicatore campionamento	2
36		Flags attività	1
37		I/O flags	1
38		Data quality flags	1
39		Numero blocchetti	1
40-43		Correzione tempo	4
44-45		Offset inizio dati	2



46-47		Offset inizio primo blocchetto	2
48-63	Vedi tab. 3 e 4	Blocchetti 1000 e 1001	16

**Tabella 1.** Struttura dell'header di 64 bytes complessivi.

I primi sei bytes rappresentano il numero incrementale di pacchetto (530469), poi troviamo le informazioni relative alla rete sismica, il nome della stazione, il codice del canale, il tempo di inizio della registrazione, il campionamento ed il numero dei "blocchetti" che seguono. Per una descrizione completa di tutti i campi si rimanda alla varia letteratura riportata nella bibliografia, mentre è importante illustrare il campo relativo al tempo di inizio della registrazione che occupa 10 byte nel seguente formato:

Posizione byte	Esempio	Nome campo	Lunghezza (bytes)
20-21	07 DB = 2011	Anno	2
22-23	00 C6 = 198 giorno giuliano corrispondente al 17 luglio	Giorno dell'anno	2
24	00	Ore	1
25	00	Minuti	1
26	00	Secondi	1
27	00	Riservato	1
28-29	06 40	.0001 secondi (0-9999)	2

**Tabella 2.** Struttura dei bytes 20-29 dell'header relativi all'orario di inizio registrazione.

Sostanzialmente il suddetto campo fornisce l'orario associato al primo campione acquisito al cambiamento del secondo che non necessariamente corrisponderà con il primo dato presente nel pacchetto; infatti bisogna tenere in considerazione che il formato *miniSeed* opera una compressione del dato dipendente dalla velocità con cui il segnale varia per cui il numero dei campioni all'interno di un pacchetto è variabile e può coprire un intervallo temporale di diversa durata, non necessariamente multiplo esatto del secondo. Per individuare esattamente il primo campione corrispondente all'orario

riportato nel precedente campo bisogna affidarsi ai suoi ultimi 2 bytes che riportano le frazioni di secondo (espresse in 1/10000 di sec) del suo offset temporale a partire dal primo dato del pacchetto che nel nostro caso è in esadecimale 0640hex ossia 1600 che moltiplicato per 1/10000 fornisce il valore di 0.16 sec. Questo vuol dire che i primi 0.16 secondi sono occupati dai dati appartenenti al secondo precedente e sapendo il campionamento (nel nostro caso di 100 c/s) si può dedurre che questi sono 16 campioni. A questo punto a dimostrazione di quanto appena detto, sappiamo dal campo 30-31 che il nostro pacchetto contiene 206 campioni e possiamo calcolare l'orario e l'offset del pacchetto successivo, con alcuni semplici passaggi:  $0.16 + 206/100 = 2.22$ , per cui l'intervallo temporale del presente pacchetto è di 2.22 secondi ed il pacchetto successivo avrà l'orario incrementato di 2 secondi (ossia 00:00:02) ed un offset di 0.22 secondi che espresso in frazioni di secondo sarà 2200 (in esadecimale 08 98) come si evince dalla figura successiva:

```
00000200h: 35 33 30 34 37 30 44 20 53 42 50 4F 20 20 20 48 ; 530470D SBPO H
00000210h: 48 5A 49 56 07 DB 00 C6 00 00 02 00 08 98 00 CE ; HZIV.Û.Æ.....I.Î
```

**Figura 2.** Primi 32 bytes del pacchetto successivo (530470). Come spiegato nel testo l'orario è 00:00:02 con un offset di 22 campioni, ossia (in virtù del campionamento noto a 100 c/s) 0.22 sec. (08 98 in esadecimale).

I blocchetti 1000 e 1001 (riquadri blu in figura 1) ci informano sul tipo di compressione utilizzata, sulla grandezza del pacchetto dati e su altri parametri descritti nelle tabelle 3 e 4.

<i>Numero</i>	<i>Esempio</i>	<i>Nome campo</i>	<i>Lunghezza (Bytes)</i>
1	03 E8	Codice blocchetto	2
2	00 38	Offset inizio prossimo blocchetto	2
3	0A	Formato di compressione	1
4	01	Big o Little Endian	1
5	09	Lunghezza pacchetto	1
6	00	Riservato	1

**Tabella 3.** Struttura del blocchetto 1000.

<i>Numero</i>	<i>Esempio</i>	<i>Campo</i>	<i>Lunghezza (Bytes)</i>
1	03 E9	Codice blocchetto	2
2	00 00	Offset inizio prossimo blocchetto	2
3	64	Timing quality	1
4	00	Microsecondi	1
5	00	Riservato	1
6	07	Numero frames	1

**Tabella 4.** Struttura del blocchetto 1001.

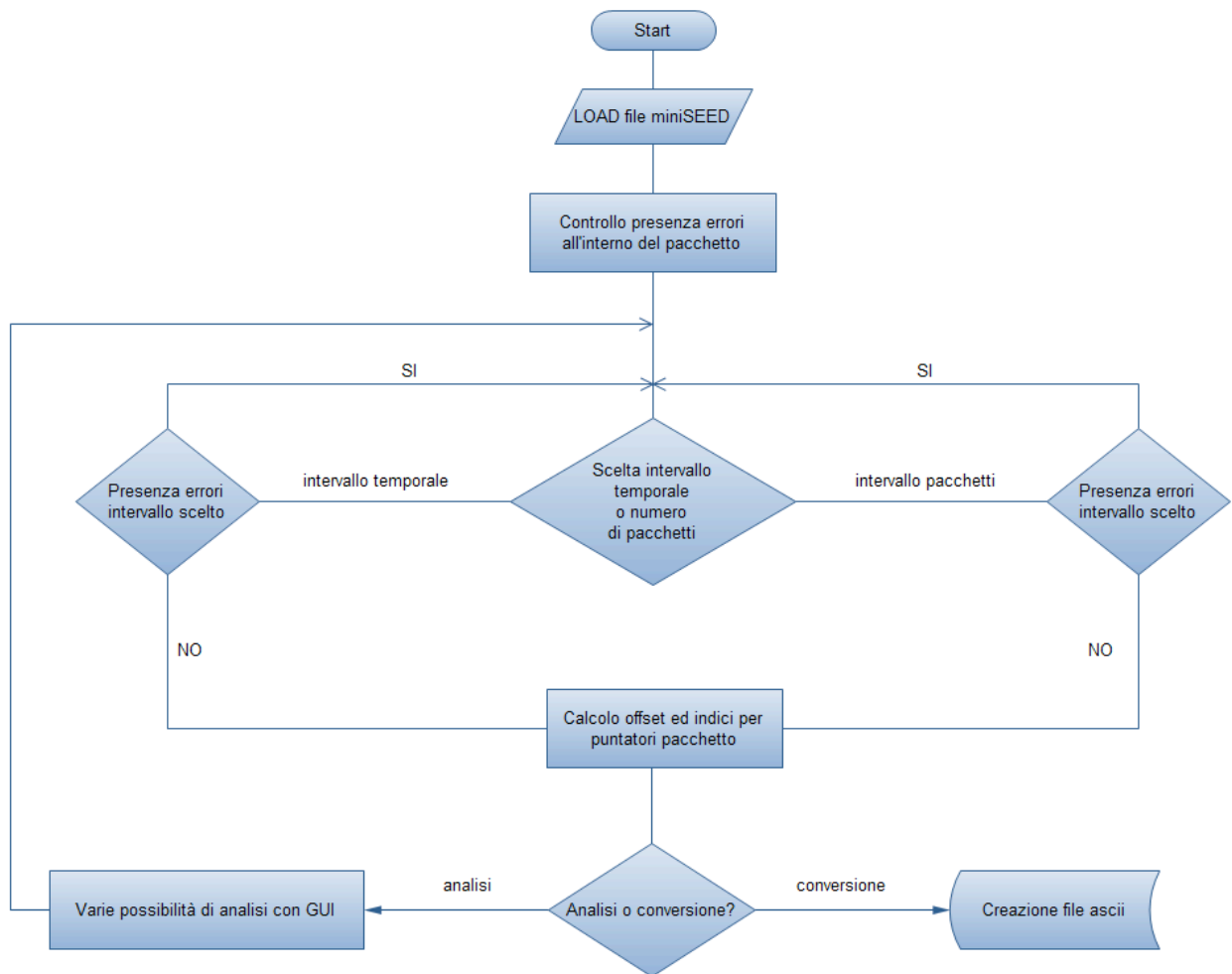
Il codice blocchetto è la conversione esadecimale del suo numero identificativo, infatti 03 E8 equivale a 1000. Di particolare rilievo, sono i campi 3 e 4, fondamentali per una corretta codifica dei dati, dal momento che esprimono il tipo di compressione (0A – Steim1 0B – Steim2) e l’ordine dei bytes ovvero *big o little endian*. Il campo 5 (09) stabilisce che l’intero pacchetto (header più dati) è di 512 bytes ossia  $2^9$ . Il riquadro verde rappresenta i dati organizzati in 7 frames di 64 bytes ciascuno (riquadri gialli) con i rispettivi bytes di codifica racchiusi nei riquadri neri, che rappresentano la vera e propria “chiave di volta” del formato *miniSeed*. Ogni frames è suddiviso a sua volta in 16 words di 4 bytes, che indicheremo come W0...W15 (ad esempio i due riquadri porpora sono W1 e W2, i due celesti rappresentano W3 e W4 mentre i riquadri neri sono tutti W0) [Rodrigo Luciano, 2004].

La codifica Steim1 introduce per come è stata concepita un determinato livello di compressione, chiaramente di tipo *lossless*, dipendente sostanzialmente dalla velocità di variazione del dato stesso. In termini semplificati si immagini di dover archiviare i seguenti quattro numeri: 500, 502, 505, 499; potrò immagazzinare i quattro valori per esteso oppure il primo valore e poi le differenze tra numeri successivi con evidente risparmio in termini di memoria occupata, ossia 500, 2, 3, -6 in virtù delle formule successive:

$$\begin{aligned}
 d_0 &= x_0 - x_{-1} \\
 d_1 &= x_1 - x_0 \\
 d_2 &= x_2 - x_1 \\
 &\dots \\
 d_n &= x_n - x_{n-1}
 \end{aligned}
 \tag{1}$$

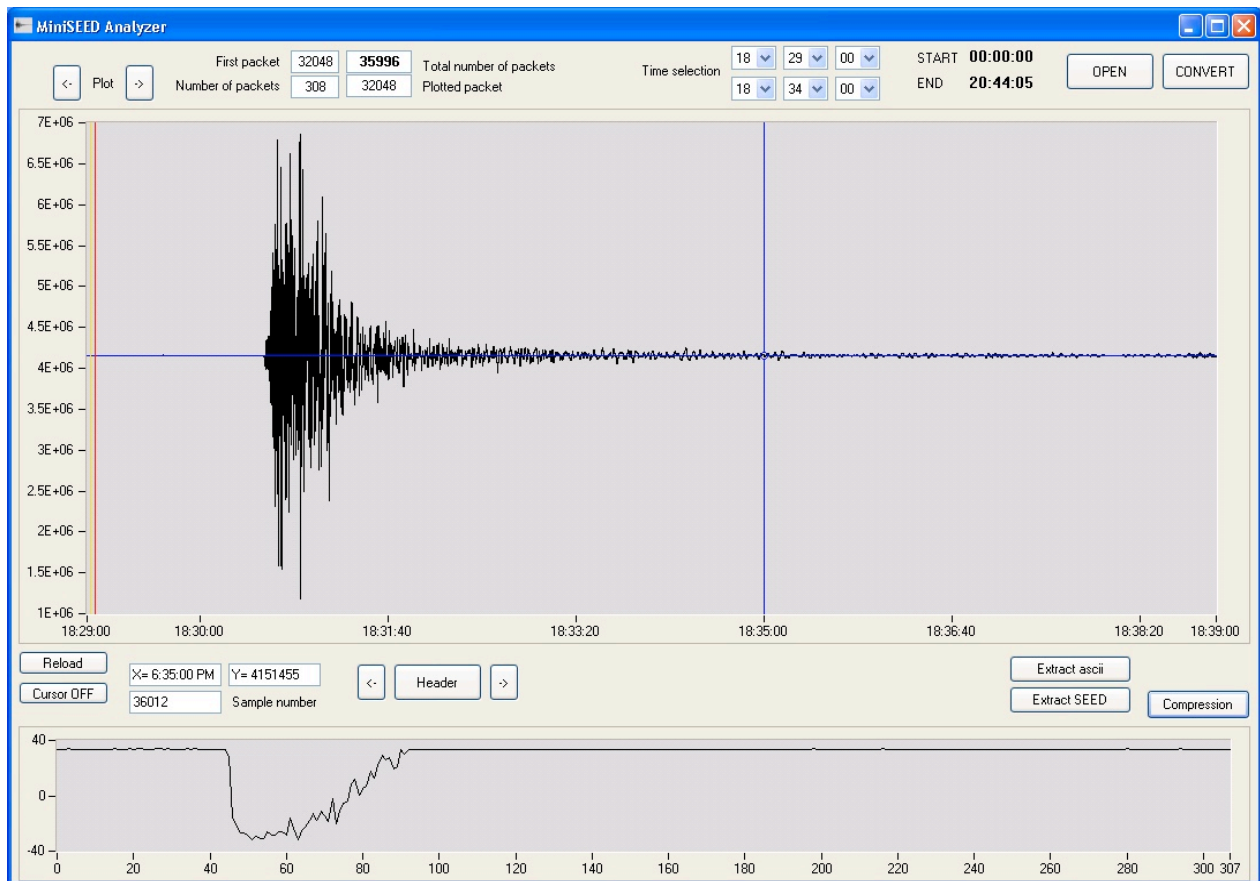
Quindi nel pacchetto dati avremo sempre il primo campione assoluto e poi una successione di differenze organizzate in vario modo a seconda della loro intensità. L’algoritmo di compressione Steim1 prevede l’immagazzinamento di un massimo di 4 differenze da 1 byte ad un minimo di una sola differenza da 4 byte all’interno di un blocco numerico da 32 bit, ossia una word del tipo W0...W15.





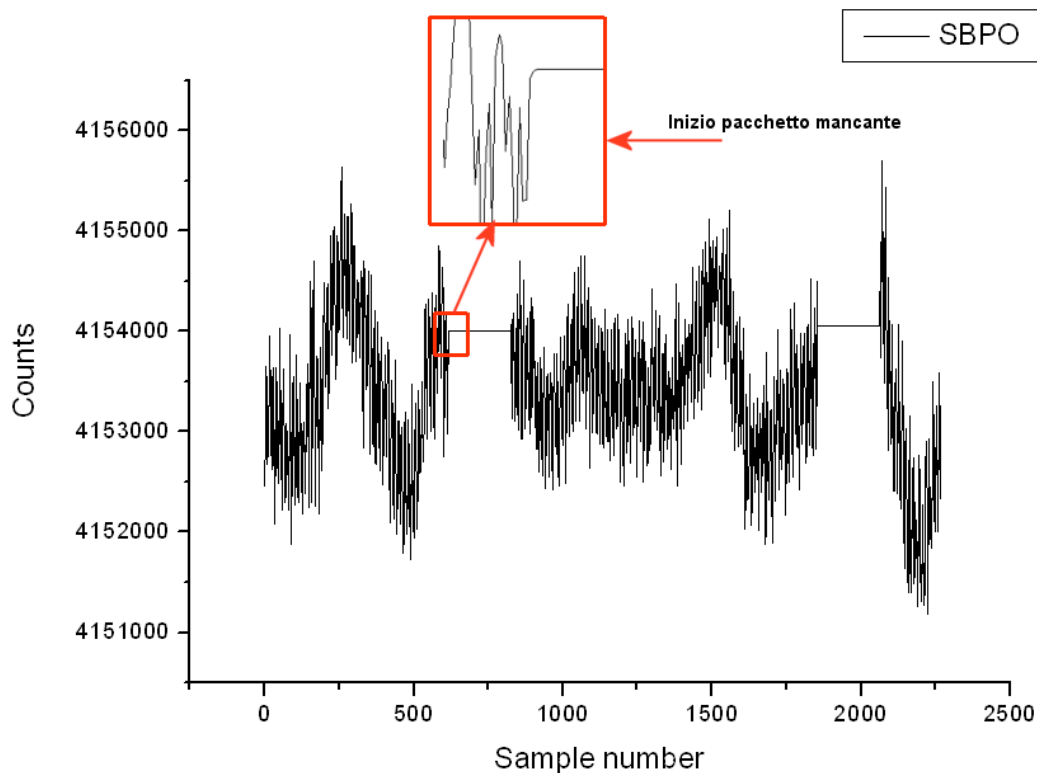
**Figura 3.** Diagramma di flusso del programma *miniSEED analyzer*.

La prima operazione richiesta dall'interfaccia utente (GUI) è il caricamento di un file *miniSeed*, operazione banalissima dal punto di vista informatico e con diverse possibilità di istruzioni per la sua realizzazione, ma molto critica dal punto di vista del tempo di esecuzione da parte della CPU. Ad esempio se si usa la funzione ***FileOpen(n,nome\_file)*** con n puntatore al singolo byte all'interno del file e quindi variabile dall'inizio alla fine del file stesso, si osserva dopo qualche secondo uno stallo della macchina soprattutto quando si tenta di aprire file di grandi dimensioni. Al contrario sfruttando il metodo ***FileReadAllBytes(nome\_file)*** che crea direttamente un vettore in ram avente le dimensioni ed il contenuto del file stesso, l'operazione di lettura dei dati diviene velocissima. Non volendo appesantire il presente lavoro con tecniche di programmazione, anche perché esulerebbe dal contesto, è importante evidenziare tuttavia, che molte scelte sono state effettuate pensando alla velocità di esecuzione e altre alla ram impiegata dal computer al fine di trovare un compromesso per la compatibilità anche su macchine un po' datate che magari eseguiranno il programma con tempi più lunghi evitando però di incorrere in problemi di *out-of-memory*. Infatti a parte l'operazione iniziale di caricamento del file, tutte le altre routine utilizzano dei vettori in memoria che ciclicamente vengono inizializzati con nuovi blocchi di dati per non occupare eccessive quantità di memoria.



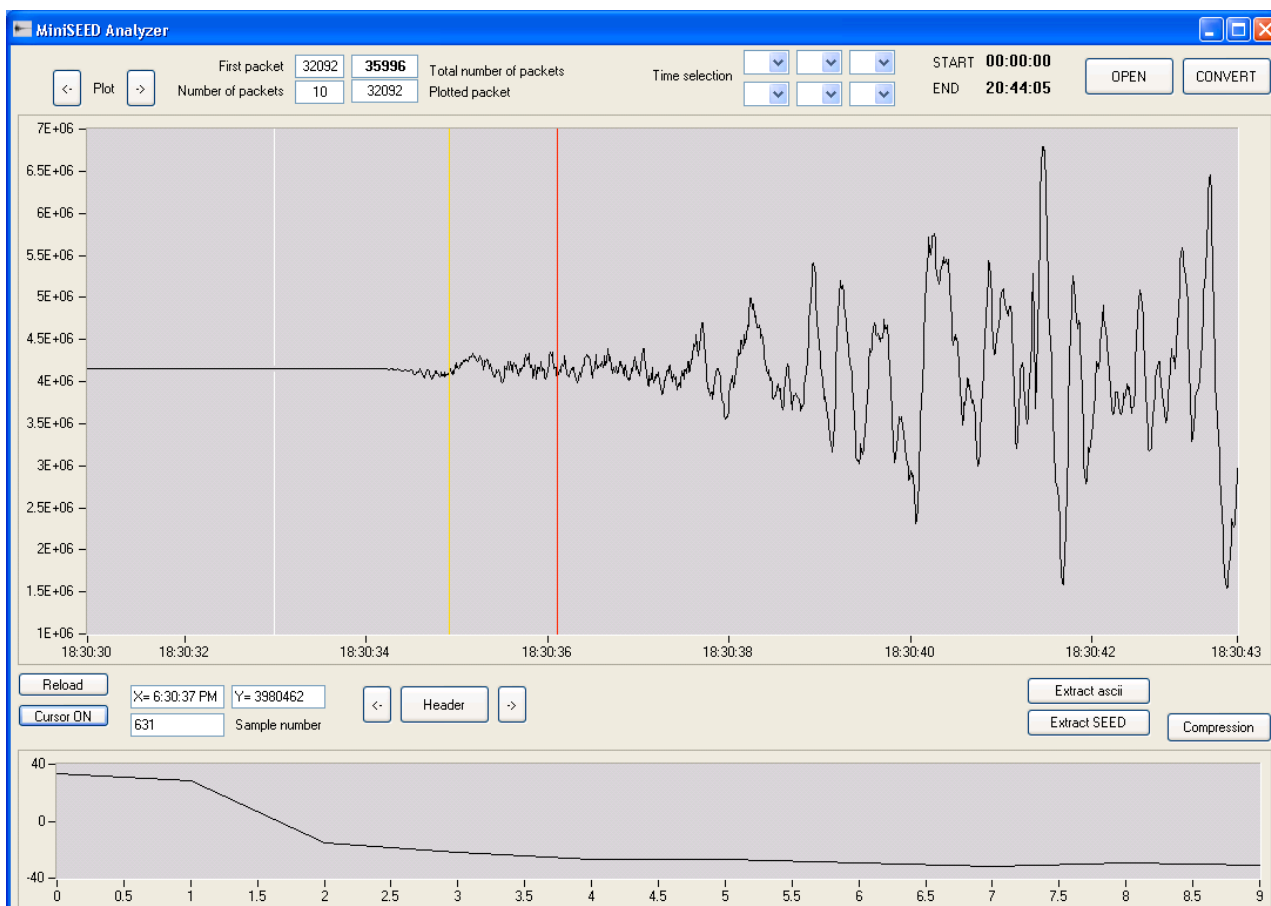
**Figura 4.** Screenshot del programma *MiniSeed Analyzer*. Sono graficati 308 pacchetti (dalle 18:29:00 alle 18:34:00 UTC) del segnale prodotto dalla stazione sismica della Rete Nazionale SBPO relativamente alla scossa di terremoto del 17 luglio 2010 del Nord Italia di magnitudo 4.7 della scala Richter. In basso viene riportato il grado di compressione insito nel formato miniseed per ciascun pacchetto.

In figura 4 si riporta la schermata principale del software. In alto a destra è visibile il pulsante “OPEN”, che una volta premuto visualizza la classica finestra di dialogo per l’apertura di un file in ambiente *Windows*. A quel punto, facendo anche riferimento alla figura 2, appena immagazzinato il contenuto del file in memoria e tenendo presente che esso è formato da una sequenza di pacchetti da 512 *bytes*, aventi tutti un *header* ed una uguale struttura dati (come precedentemente illustrato), si procede al controllo progressivo del numero di pacchetto, individuando delle eventuali discontinuità; qualora ce ne fossero vengono memorizzate le loro posizioni in riferimento ad un indice che si incrementa ciclicamente. Terminata la memorizzazione dei dati si conosceranno, qualora ce ne fossero, le posizioni esatte dei pacchetti mancanti, che saranno riempiti nell’eventuale fase di conversione, premendo “CONVERT”, con l’ultimo valore valido prima del gap. In pratica, volendo fornire un esempio, se mancasse il pacchetto 562368, l’algoritmo di controllo individua gli estremi temporali di inizio e fine e in virtù del campionamento ne determina il numero di campioni mancanti  $n$ ; a questo punto scriverà  $n$  volte il valore dell’ultimo dato valido convertito ed appartenente al pacchetto 562367. La figura 5 mostra la conversione di un file con due pacchetti mancanti.



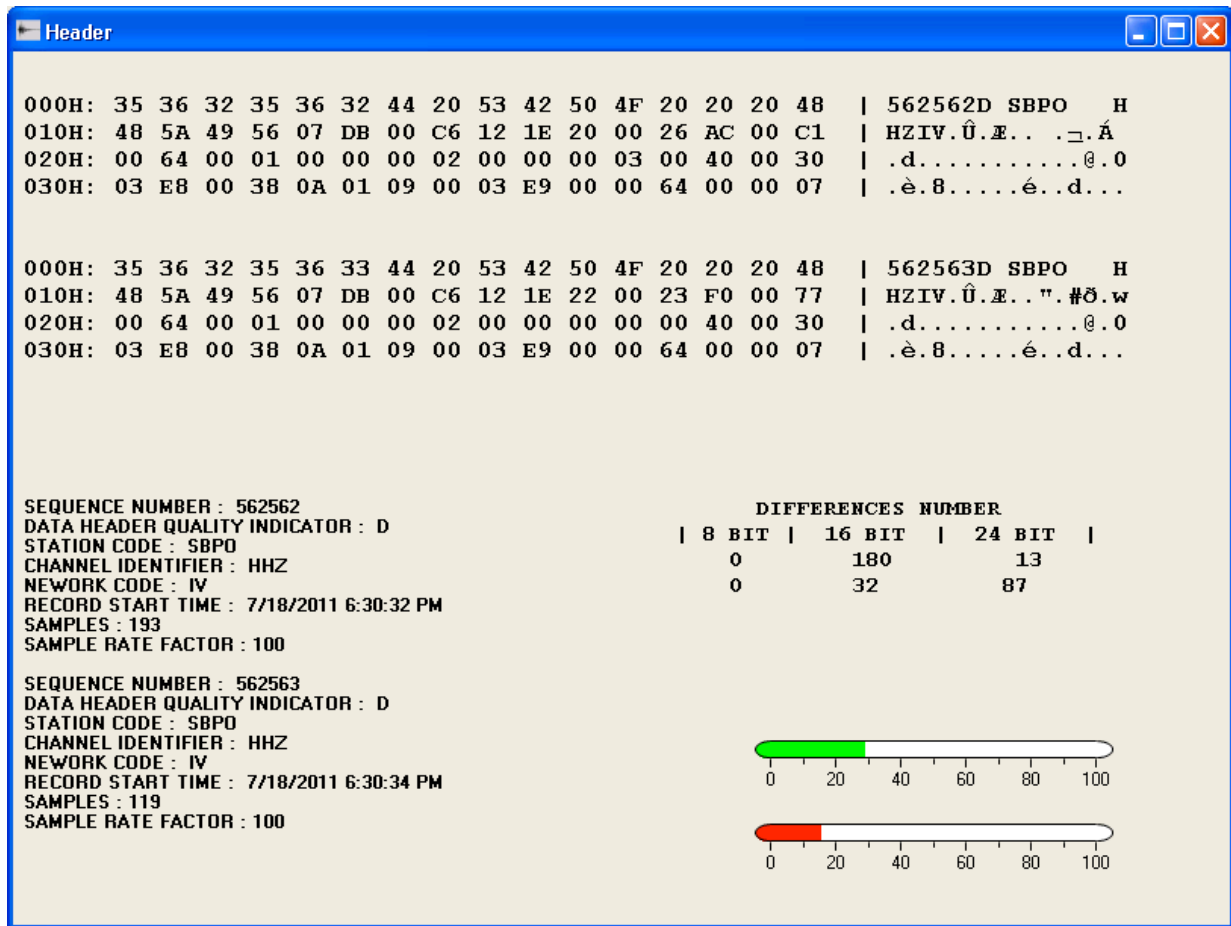
**Figura 5.** Rappresentazione grafica tramite il software Origin, di dati ascii convertiti con *miniSeed analyzer* da un pacchetto miniseed avente dei dati mancanti. L'ultimo campione valido viene ripetuto al posto delle informazioni mancanti.

Facendo una piccola digressione alla fase di caricamento del file, se l'utente avesse scelto di operare l'analisi dei dati, invece della conversione scegliendo un intervallo che comprendesse degli errori il programma avverte con un messaggio informativo, ma non tiene conto in nessun modo dei valori mancanti. In altri termini non vengono rappresentati valori fittizi, ma solo campioni fisicamente presenti nel pacchetto. Come si può notare dal diagramma di flusso, il programma offre la possibilità di lavorare con il riferimento temporale o con quello del numero di pacchetto. In alto, al centro della figura 4 si possono vedere l'orario di inizio e fine registrazione dell'intero file, a sinistra di tale zona ci sono le finestre di dialogo per la selezione temporale relativa all'estrazione di porzioni di file. Se l'utente non sceglie nessun orario il programma prende automaticamente in considerazione una selezione basata sul puntamento al numero di pacchetto e come quantità di dati il numero di pacchetti desiderati. Ad esempio il default prevede l'inizio dell'analisi a partire dal pacchetto 0 per un intervallo di un singolo pacchetto. Con i due pulsanti in alto a sinistra l'utente può navigare nella visualizzazione dei pacchetti ad intervalli prestabiliti all'inizio o modificabili in qualsiasi momento fosse necessario. Se si sceglie un intervallo che coinvolge fino ad un massimo di 1000 pacchetti, vengono visualizzati nell'area del grafico delle linee verticali di diverso colore, come mostrato nella figura successiva.



**Figura 6.** Forma d'onda di 10 pacchetti dell'inizio del sisma. Le tre linee verticali colorate individuano due pacchetti successivi da 512 byte. La differente distanza indica un diverso numero di campioni rappresentati e conseguentemente un diverso rapporto di compressione.

Le suddette linee individuano l'inizio e la fine di due pacchetti successivi, ossia in altri termini la loro distanza è proporzionale al numero di campioni inclusi nel pacchetto. Con i due pulsanti in basso a sinistra si possono variare le loro posizioni all'interno del pacchetto. Dalla figura 6 si può immediatamente comprendere come la porzione di segnale tra la linea bianca e quella gialla sia più compressa di quella tra la linea gialla e quella rossa, ricordando quanto appena detto sulla relazione tra distanza e numero di campioni. Agendo sul pulsante "Compression" viene graficato il rapporto di compressione per ogni pacchetto dell'intervallo scelto, nel caso in figura si tratta di 10 pacchetti a partire dal numero 32092, ed il suo andamento è conforme a quanto suddetto. Il pulsante "Header" mostra la finestra in figura 7 in cui è possibile ricevere tutte le informazioni relative all'header dei due pacchetti compresi tra le tre linee, che sono adiacenti e cronologicamente sequenziali. Viene mostrato il *dump* dei *byte* in formato esadecimale e laddove possibile l'interpretazione in *ascii*. Inoltre i vari campi dell'header vengono interpretati ed in basso se ne mostra il significato. In aggiunta alle informazioni contenute nell'header viene riportato il conteggio con rispettiva tipologia delle differenze che compongono la codifica *miniSeed*. Ritornando ai dati racchiusi dai cursori nella figura 6 si può vedere come il secondo pacchetto abbia un rapporto di compressione molto svantaggioso ed addirittura superiore al pacchetto non compresso. I due indicatori esprimono infatti la compressione sotto forma di percentuale, in relazione ad un formato dati a 24 bit. Nel caso in cui il formato *miniSeed* occupi più memoria rispetto alla non compressione il colore diviene rosso e viene indicata la percentuale di memoria occupata in eccesso. Ad esempio il primo pacchetto offre una compressione di circa il 30%, mentre il secondo supera del 15% il formato non compresso; infatti il numero di campioni nel secondo pacchetto è sceso, rispetto al primo, da 193 a 119 o in termini di differenze da 180 differenze a 16 bit si passa a 32 e da 13 a 24 bit si sale a 87, quindi il segnale comincia a variare nel tempo con più velocità.

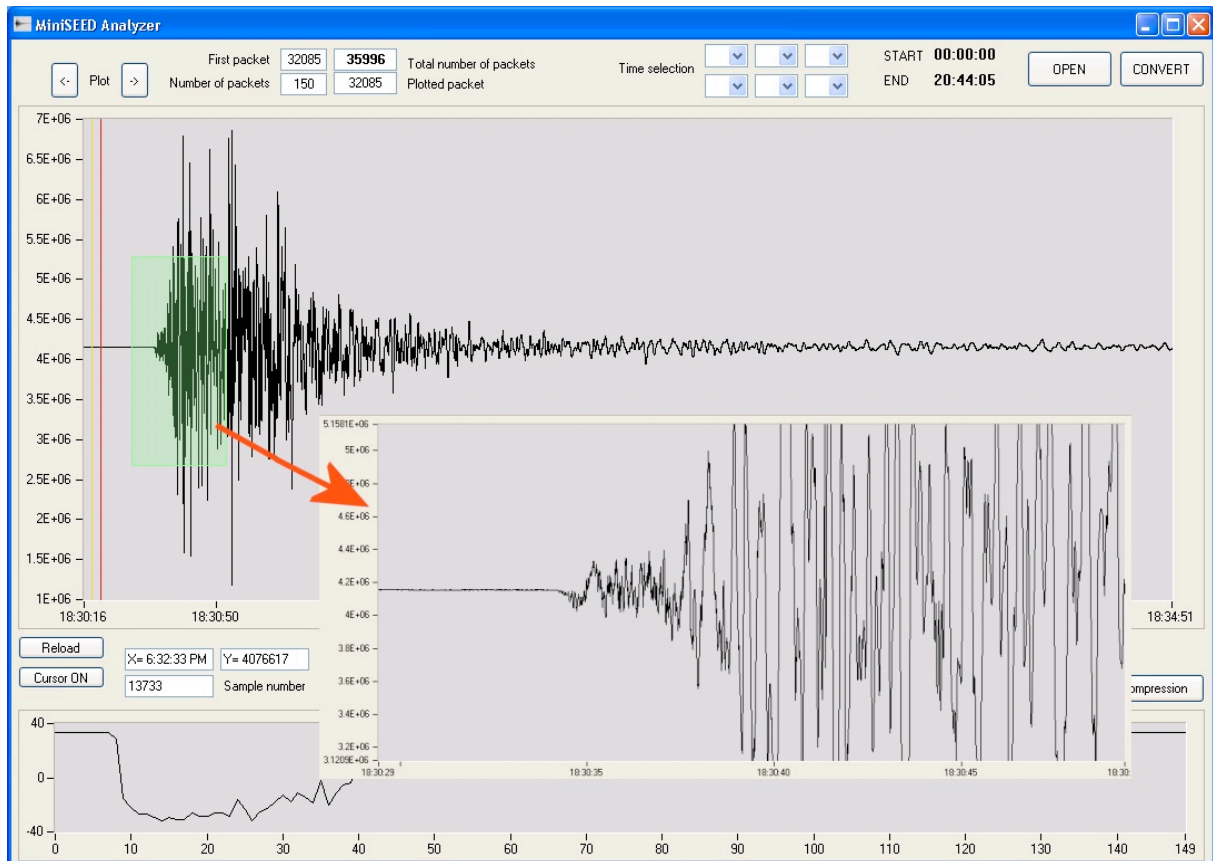


**Figura 7.** Finestra di *miniSeed analyzer* dedicata alle informazione racchiuse nell’header da 64 bytes di due pacchetti temporalmente adiacenti.

L’utente operando un doppio click del mouse in qualsiasi area del grafico, attiva la modalità zoom come visibile in figura 8, con selezione della zona da ingrandire. Tale azione è ripetibile quante volte si vuole, in accordo con il grado di ingrandimento che si vuole raggiungere. È sempre possibile tornare indietro con il tasto “Reload”, che ripristina la situazione precedente, fino a tornare al segnale originale se premuto ripetutamente. In ogni fase di visualizzazione si può sfruttare un cursore vincolato alla curva del grafico e leggere i valori delle ascisse e ordinate negli appositi spazi a destra del pulsante “Cursor on”. Le ascisse esprimono l’orario del segnale, mentre in basso si riporta il numero del campione considerando come numero 1 il primo dato dell’area visualizzata. La porzione di segnale visualizzato può essere trasferita su file, tramite i pulsanti “Extract ascii” ed “Extract SEED” nel formato desiderato.

Se si sceglie di navigare tra i dati avanzando di un pacchetto alla volta, scrivendo 1 nel campo “Number of packets”, il programma fornirà nuovi strumenti di analisi, dal momento che non avrebbe avuto senso memorizzare grandi quantità di informazioni, per ampi intervalli di dati, le quali avrebbero perso leggibilità creando inevitabilmente confusione. Al contrario analizzando in modo dettagliato il singolo pacchetto ed eventualmente quelli adiacenti si può facilmente condurre un’analisi efficace e puntuale. Nella finestra mostrata in figura 5 comparirà il pulsante “View differences” mentre nel pannello principale saranno visibili dei campi che permetteranno l’introduzione di alcuni parametri caratteristici dell’acquisitore o del sensore impiegato al fine di poter esprimere nelle corrette unità di misura, le grandezze fisiche in questione attraverso il pulsante “Replot” (vedi fig. 9).



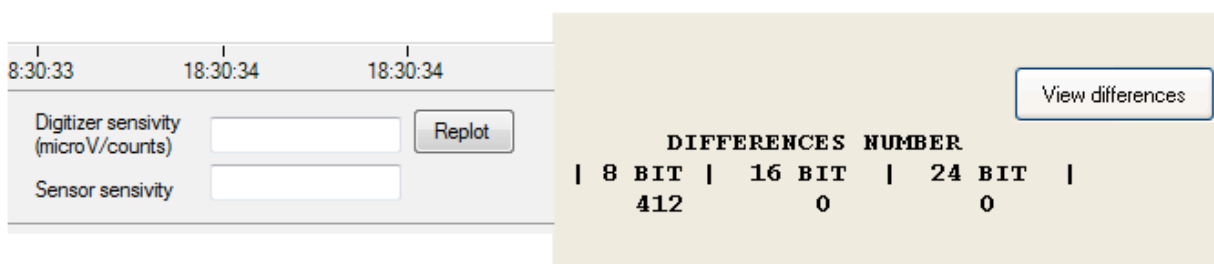


**Figura 8.** Funzione di zoom. Con il tasto “Reload” si può ripristinare il precedente grado di ingrandimento.

Cliccando su “View differences” si aprirà una nuova finestra, dove verranno riportati tutti i valori numerici delle differenze costituenti la codifica *miniSeed*, raggruppati a seconda della loro grandezza in bytes. Inoltre verrà data una rappresentazione grafica di tali valori. Facendo riferimento al concetto di derivata prima di una funzione si può facilmente intuire come l’insieme delle differenze rappresenti, a meno di fattori moltiplicativi, la derivata del segnale codificato. Essendo infatti la derivata prima di una funzione  $x$  data da:

$$x'_n = (x_n - x_{n-1})/h \quad (2)$$

e facendo riferimento all’ultima formula (1) di pagina 9 si capisce come il numeratore della (2) è identificabile con la differenza tra due campioni successivi, chiave della codifica *miniSeed*, mentre il denominatore  $h$  è dato dal passo di campionamento ed il suo reciproco  $1/h$  è il fattore moltiplicativo appena accennato.



**Figura 9.** Pulsanti e campi aggiuntivi relativi all’analisi del singolo pacchetto.

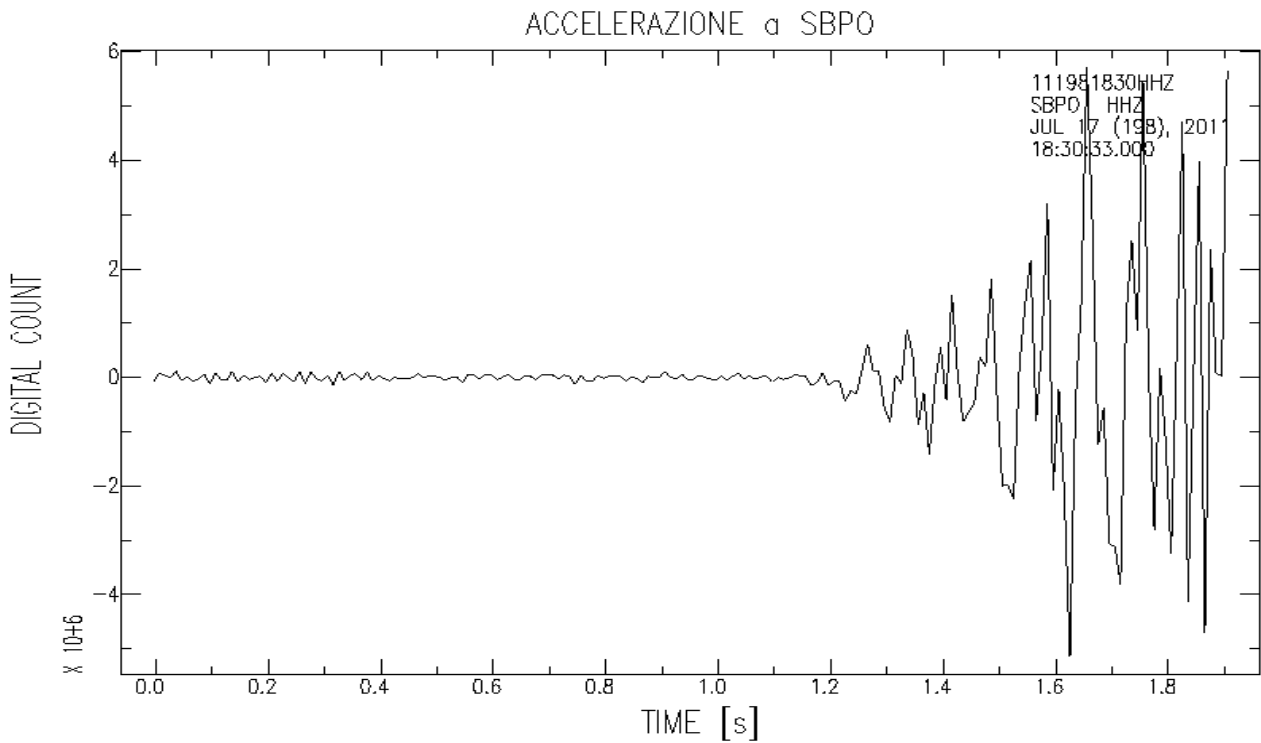
In altre parole si può concludere che nella codifica *miniSeed* di un segnale, implicitamente si ha anche la sua derivata prima, a patto di prendere in considerazione solo le differenze senza l'applicazione delle formule (1) di codifica del segnale.

La figura 10 rappresenta un pacchetto proveniente da un velocimetro (sensore Trillum 40 sec.), relativo all'inizio di un terremoto. Nella figura 11 è illustrata la sua derivata prima, eseguita con il programma SAC in ambiente Linux con l'intenzione di ricevere informazioni sull'accelerazione del terreno e come si può osservare dalla figura 12, che è il grafico delle sole differenze (ottenuto tramite "View differences") di codifica del segnale di figura 10, si ottiene esattamente lo stesso risultato a meno del fattore moltiplicativo che in questo caso è  $1/0.01 = 100$ .

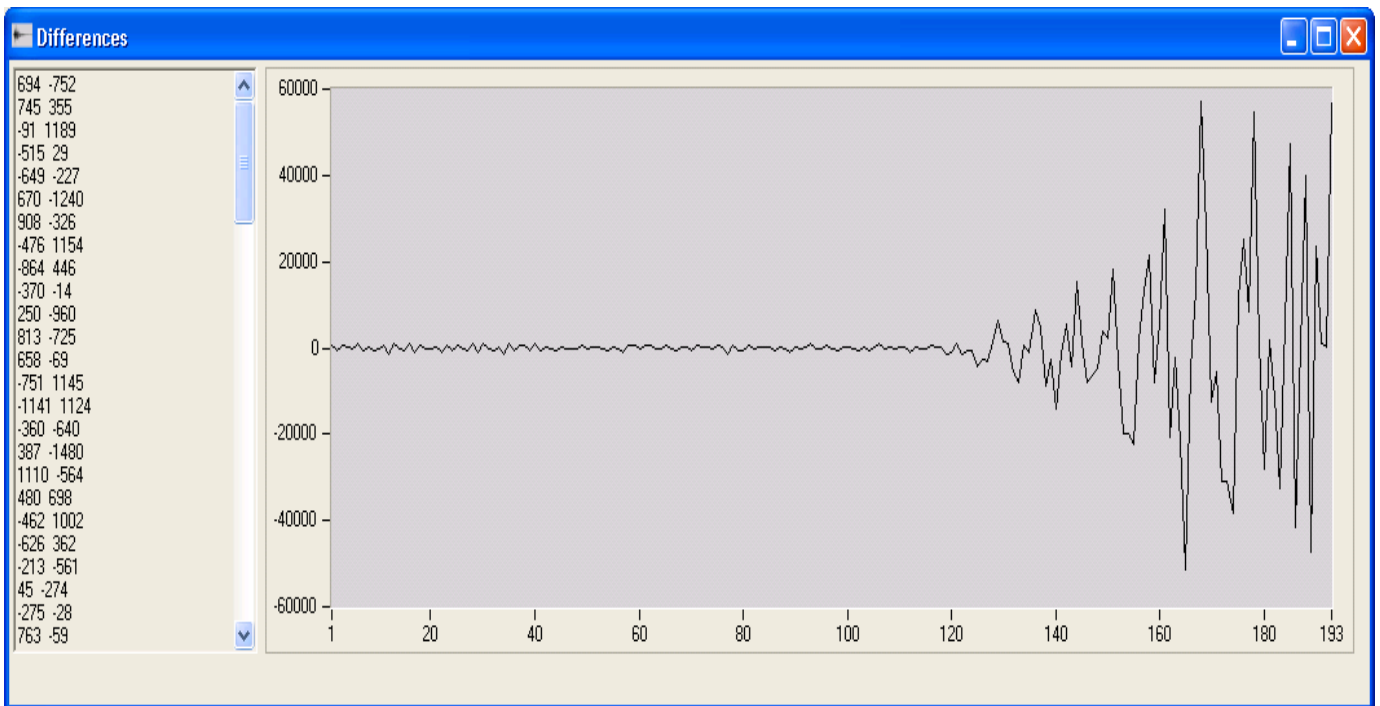
La suddetta considerazione, che a tutta prima potrebbe risultare elementare, in realtà può essere adoperata dagli sviluppatori di software per l'analisi della derivata dei segnali sismici, a "costo zero", essendo implicita nel tipo di codifica *miniSeed*.



**Figura 10.** Pacchetto numero 32093 relativo all'inizio del sisma.



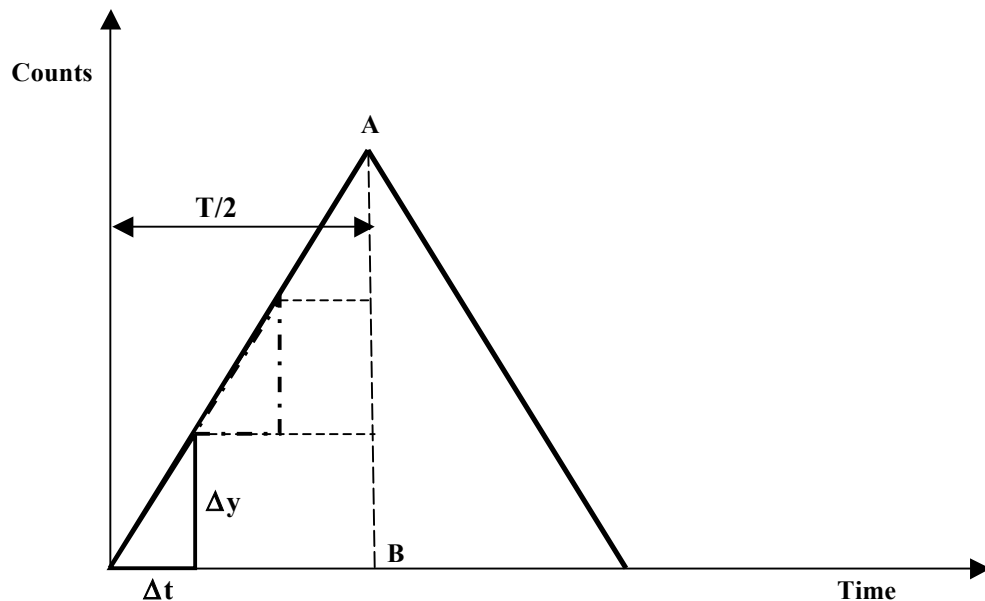
**Figura 11.** Accelerazione ricavata tramite l'operazione di derivazione con il programma SAC in ambiente Linux, relativa al pacchetto velocimetrico della figura 9.



**Figura 12.** Differenze della rappresentazione miniseed del pacchetto riportato in figura 9 ed ottenuto con il pulsante "View differences". Come si può osservare il risultato è coincidente con il calcolo della derivata ottenuta con il software SAC della precedente figura.

### 3. Considerazioni sul formato *miniSeed*

Nel presente paragrafo si analizzerà il comportamento del formato *miniSeed* e dell'efficienza di compressione relativamente ad un segnale non sismico. È stata generata con un generatore di segnali un'onda triangolare con ampiezza di 1.5 volts e frequenza tale da poter essere codificata in *miniSeed* con tutte differenze da 16 bit, secondo il ragionamento successivo. Tale onda è stata acquisita con un acquisitore GAIA2 con fondo scala di 10 vpp ossia con sensibilità di  $0.7947 \mu\text{V}/\text{counts}$ . Si prenda in considerazione un'onda triangolare come quella rappresentata nella figura 13. Vogliamo determinare la frequenza massima alla quale il segnale è codificabile in *miniSeed* con tutte differenze da 16 bit.



**Figura 13.** Rappresentazione geometrica di una forma d'onda triangolare.

La suddetta condizione impone:

$$\Delta y < 2^{16}/2 \quad (3)$$

In virtù del fatto che le differenze sono espresse con numeri binari di tipo signed (con segno) per cui i valori positivi si dimezzano.

Dalla geometria del segnale si ricava che:

$$(AB/\Delta y) * \Delta t = T/2 \quad (4)$$

dove  $\Delta t$  è il passo di campionamento e nel nostro caso vale 0.01 sec

Ossia:

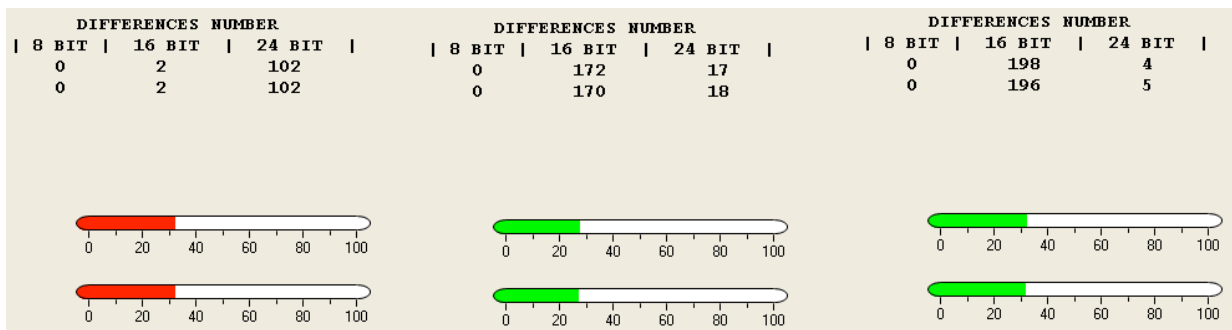
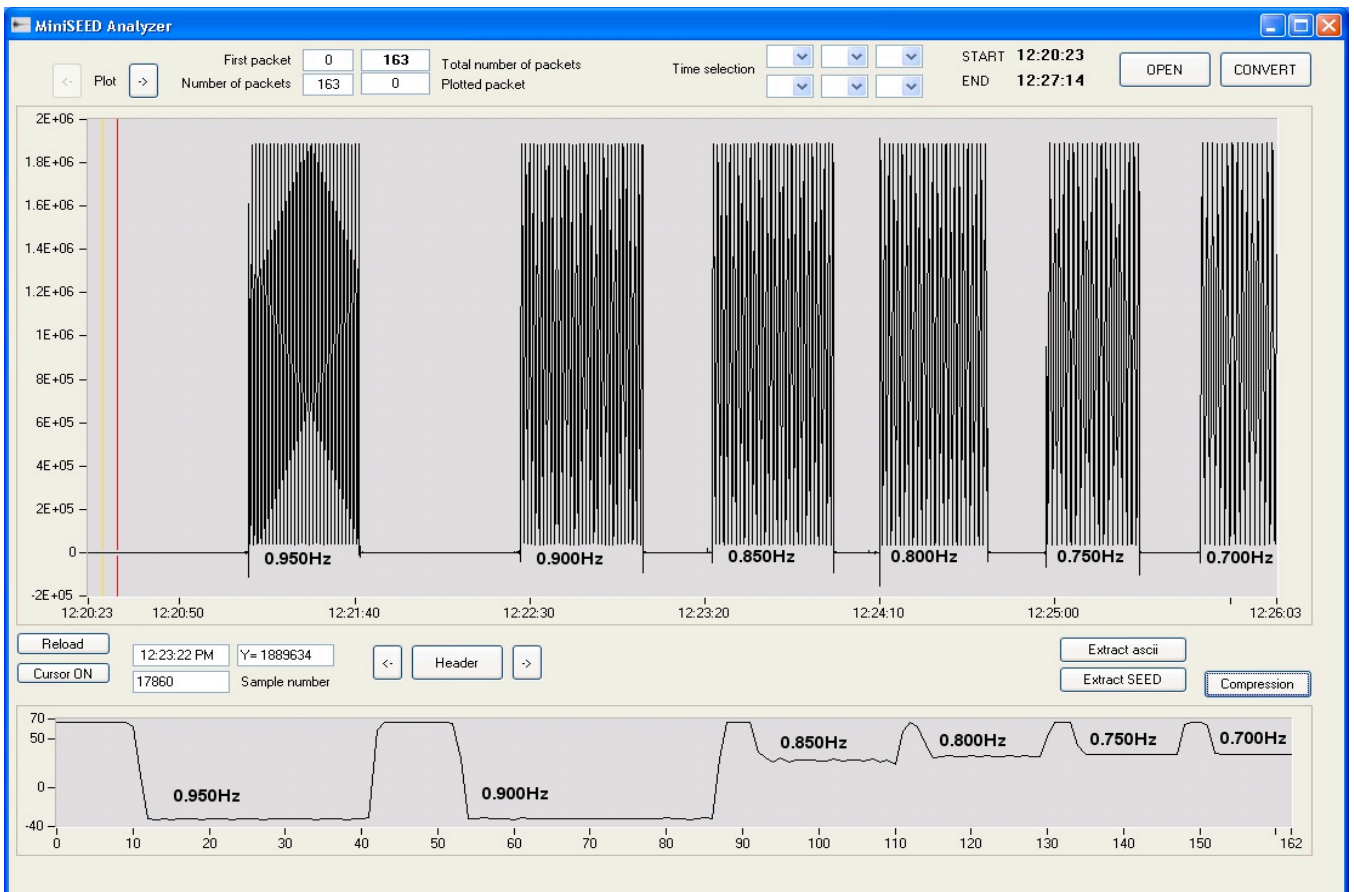
$$T = (2AB * \Delta t) / \Delta y \quad (5)$$

Per cui si otterranno tutte differenze a 16 bit per  $T > (2AB * \Delta t) / \Delta y$  o ragionando in frequenza per  $f < \Delta y / 2AB * \Delta t$ .

Passando ai numeri si ha  $\Delta y = 32768 \text{ AB} = (1.5\text{V} * 0.7947 \mu\text{V}/\text{counts}) = 1.8875 * 10^6$

Quindi  $f < 32768/2 * 1.8875 * 10^6 * 0.01$  da cui  $f < 0.868 \text{ Hz}$ .

Nella figura 14 si riporta il caso di diversi segnali triangolari con frequenze variabili tra 0.95 Hz e 0.70 Hz e successivamente le informazioni sull'entità delle differenze. Come si evince dalle illustrazioni, in concomitanza di quanto suddetto, a 0.85Hz le differenze a 16 bit sono la quasi totalità essendo comunque il calcolo svolto con dei valori teorici.



**Figura 14.** Andamento della compressione in funzione della frequenza. In basso si riporta la situazione dettagliata della compressione e del numero di differenze, relativamente ai casi 0.9, 0.85 e 0.8 Hz.

Come appena dimostrato la compressione comincia ad essere molto efficiente per segnali caratterizzati da bassi valori in frequenza. Si pensi che per ottenere solo differenze ad 8 bit bisogna scendere a circa 7

mHz. Per cui tale algoritmo di compressione potrebbe apparire poco efficace se applicato a segnali prodotti in laboratorio, ma in realtà a contribuire al suo successo è la tipologia del segnale sismico, che come dimostrato per circa il 99% del tempo di solito è composto da solo rumore sismico, che indipendentemente dalla sua frequenza ha un valore in modulo molto basso e quasi sempre esprimibile con differenze da 8 bit.

#### **4. Sviluppi futuri e conclusioni**

Con il presente lavoro si è cercato di fornire un utile strumento per l'analisi del formato *miniSeed* nelle sue peculiarità, con l'intenzione di poter rapidamente eseguire un controllo sullo streaming di dati prodotti da un acquirente sismico o da un software che producesse campioni in tale formato. Come si è potuto constatare il maggior numero di strumenti il programma li offre quando si esplora il segnale prendendo in considerazione un pacchetto per volta, per ovvi fini di risparmio di memoria e maggiore leggibilità delle informazioni. Inoltre si è data la dimostrazione che ogni segnale espresso con la codifica *miniSeed* implicitamente reca con sé l'informazione sulla sua derivata prima in virtù di come è stata concepita la codifica stessa, implicazione molto interessante per chi deve sviluppare routine in programmi di analisi di segnali sismici.

Al momento *miniSEED analyzer* funziona solo con la codifica Steim1, anche se è in grado di rilevare la Steim2 non può decodificarla. Un futuro sviluppo prevede una maggiore generalizzazione nell'interpretazione dei possibili tipi di codifica anche con pacchetto di dimensioni diverse da 512 byte ed la possibilità di visualizzare real-time un flusso dati in *miniSeed* proveniente dalla rete, tramite il protocollo TCP/IP, e contemporaneamente ricevere informazioni sulla sua derivata sempre in tempo reale.

#### **Bibliografia**

- International Federation of Digital Seismograph Networks Incorporated Research Institutions for Seismology - United States Geological Survey (2012). *SEED Reference Manual, Seed Format Version 2.4, August 2012*.
- IASPEI, *New Manual of Seismological Observatory Practice (NMSOP)*.
- Rodrigo Luciano Pereira Canabrava (2004). *Norwegian National Seismic Network, A toolbox for reading SEED and MiniSEED and writing MiniSEED*, Technical Report No. 18, December 2004.

#### **Sitografia**

- <http://www.seiscomp3.org/wiki/doc/applications/seedlink>  
<http://www.iris.edu/manuals/>

#### **Ringraziamenti**

Si ringrazia il Dott. Paolo Casale per aver effettuato l'analisi di un pacchetto dati provenienti da un velocimetro ricavandone la derivata prima temporale attraverso il software SAC in ambiente Unix riportata nella figura 11.

**Coordinamento editoriale e impaginazione**

Centro Editoriale Nazionale | INGV

**Progetto grafico e redazionale**

Daniela Riposati | Laboratorio Grafica e Immagini | INGV

© 2014 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

**<http://www.ingv.it>**



**Istituto Nazionale di Geofisica e Vulcanologia**