# Gautebøye - Processing

Gaute Hope (gaute.hope@student.uib.no), 21.08.2012, Revision 1

## Introduction

This document describes the simple processing tools as well as some notes for how to use the data with SEISAN or MATLAB.

## 1 Tools

### 1.1 dtttomseed

### 1.2 datreader

Reads IND and DAT files and can, among other things, convert to DTT. Typical invocation for converting a large number of DAT files to DTT (*bash*).

    for i in *.IND; do datreader -s -d >
${i/IND/DTT}; done

### 1.3 dttfix

### 1.4 fakedtt

### 1.5 mschangesource

## 2 SEISAN

### 2.1 Configuration

You have to increase the plot resolution of mulplt (MULPLT.DEF) to something like 60000 (screen resolution and hardcopy resolution), as well as update the frequency band of the spectral analysis to at least 125 Hz. More might be required. There is 40960 points in each DAT or DTT file.

### 2.2 Conversion

Use *dtttomseed* or *dattomseed* to convert to miniSEED, update the source file to change location, station is given as first argument. I have used SHZ as channel code for the hydrophone.

### 2.3 Test

Use the program *fakedtt.py* to create a DTT file with a sine wave and try to convert and plot it to see if everything works.

### 2.4 Importing to SEISAN

1. Create a work directory, copy the data and index files for the events to it.

2. Use dtttomseed or dattomseed with the first argument being station name, BUO1 to BUO5 and after that a list of sequences of the ids of the files.

3. A mseed file is created in the current directory.

4. Use mscut to split the file up in hourly intervals.

5. Create a list of the final files with dirf.

6. Use autoreg to add the events to the database. Copy the WAV files and select create new IDs if they conflict with existing events.

7. Use associ to merge the events with the existing events in the database. Delete merged events.

### 2.5 Continuous database setup

1. Set up a database for the experiment, e.g. GAK1

2. Import events hourly as described above, one event for each hour.

3. Set up SEISAN.DEF to know GAK1 as a continuous database.

4. Set a new database for registered and extracted events, e.g. GAK1E.

5. Work on GAK1 and register ('p' in mulplt) to create an event in GAK1E when something interesting is found.

6. GAK1 can be used for noise measurements etc.

7. Possibly create an intermittent stop with sanitized events (fixed time, etc..)

## 3 MATLAB