



## European volcanological supersite in Iceland: a monitoring system and network for the future

### Report

#### D6.4 - Near-real-time correlation of high-rate geodetic data and seismic data from target volcanoes

Work Package:	<i>Imminent eruptive activity, eruption onset and early warning</i>		
Work Package number:	6		
Deliverable:	<i>Near-real-time correlation of high-rate geodetic data and seismic data from target volcanoes</i>		
Deliverable number:	6.4		
Type of Activity:	RTD		
Responsible activity leader:	<i>Kristín Vogfjörð</i>		
Responsible participant:	<i>Icelandic Meteorological Office (IMO)</i>		
Authors:	<i>Matthew J. Roberts (IMO)</i>		

Type of Deliverable:	<i>Report</i>	<input type="checkbox"/>	<i>Demonstrator</i>	<input type="checkbox"/>
	<i>Prototype</i>	<input checked="" type="checkbox"/>	<i>Other</i>	<input type="checkbox"/>
Dissemination level:	<i>Public</i>	<input checked="" type="checkbox"/>	<i>Restricted Designated Group</i>	<input type="checkbox"/>
	<i>Prog. Participants (FP7)</i>	<input type="checkbox"/>	<i>Confidential (consortium)</i>	<input type="checkbox"/>

**Seventh Framework Programme**  
**EC project number: 308377**



## Summary

Deliverable 6.4 demonstrates how high-rate geodetic and seismic data can be integrated in near-real-time for volcanic monitoring purposes. The work takes advantage of the latest developments in data streaming and graphical visualisation. It addresses a fundamental requirement to minimise the time-delay between raw data collection and the availability of useable, multidisciplinary monitoring results. Data are processed into easily understandable visual displays, enabling monitoring scientists and hazard managers to evaluate ongoing hazards in near-real-time. The freely available R programming language was used to process and visualise various types of geophysical data. A processing environment was developed to minimise the time-delay between data collection and the display of useable results. Two combined online displays were developed, one for continuous monitoring of the Hekla volcano and the other for following caldera subsidence at the Bárðarbunga volcano between September 2014 and February 2015. The results show how automated, rapidly available data-plots can be used to follow escalating levels of volcanic unrest in near-real-time.

## Introduction

This report outlines the results of task 6.8 (sub-task 2) within work-package six, which focuses on near-real-time integration of geodetic and seismic data for improved monitoring of volcanic hazards. In the context of this report, near-real-time pertains to the delay between the occurrence of an event and the availability of human-readable data products. Delays could be caused by data transfer, automated data processing or any other hindrance that prolongs the delivery of meaningful information.

For effective geophysical monitoring and assessment of volcanic unrest, rapid visualisation of monitoring data, in both raw and processed formats, is paramount. Such an approach allows time-dependent hazard assessments to be made in phase with escalating levels of volcanic unrest (Tilling, 1989, 2008; Dzurisin, 2003). The 2010 summit eruption of Eyjafjallajökull, Iceland, is a prime example of how temporally continuous GPS and seismic data were used to follow the pre-eruptive deformation of the volcano, as well as the evolution of the eventual eruption (Sigmundsson *et al.*, 2010). For volcanoes exhibiting hardly any pre-eruptive signals, the case for rapid, automated visualisation of geophysical data is even more important. A case in point is Hekla, Iceland, which has erupted twice since 1991; on both occasions the time between apparent quiescence and an explosive eruption has been less than an hour (Linde *et al.*, 1993). For the 2000 eruption of Hekla, the early detection of magma movement via a nearby borehole strainmeter, together with detection of increased micro-seismicity, resulted in a public warning ~20 minutes ahead of the breakout of the eruption (Agustsson *et al.*, 2000) (Figures 1 and 2).

The deliverable's overall purpose is to develop an operationally reliable product that can be used by natural hazard specialists at the Icelandic Meteorological Office (IMO). The work deviated from initial plans due to volcanic unrest at Bárðarbunga in August 2014, which culminated in the Holuhraun eruption, lasting six months (Sigmundsson *et al.*, 2015). The original target volcanoes were Grímsvötn, Hekla and Katla, but events at Bárðarbunga allowed the development work to be expanded. Moreover, several automated plots have featured on IMO's website as publicly accessible resources during the Holuhraun eruption, thereby maximising the outreach potential of the work.

The following section highlights the modern-day predicament of too much incoming data and not enough useable results. This is followed by a description of the methods used in this study. The remainder of the report highlights several novel methods for integrating and displaying geophysical monitoring data.

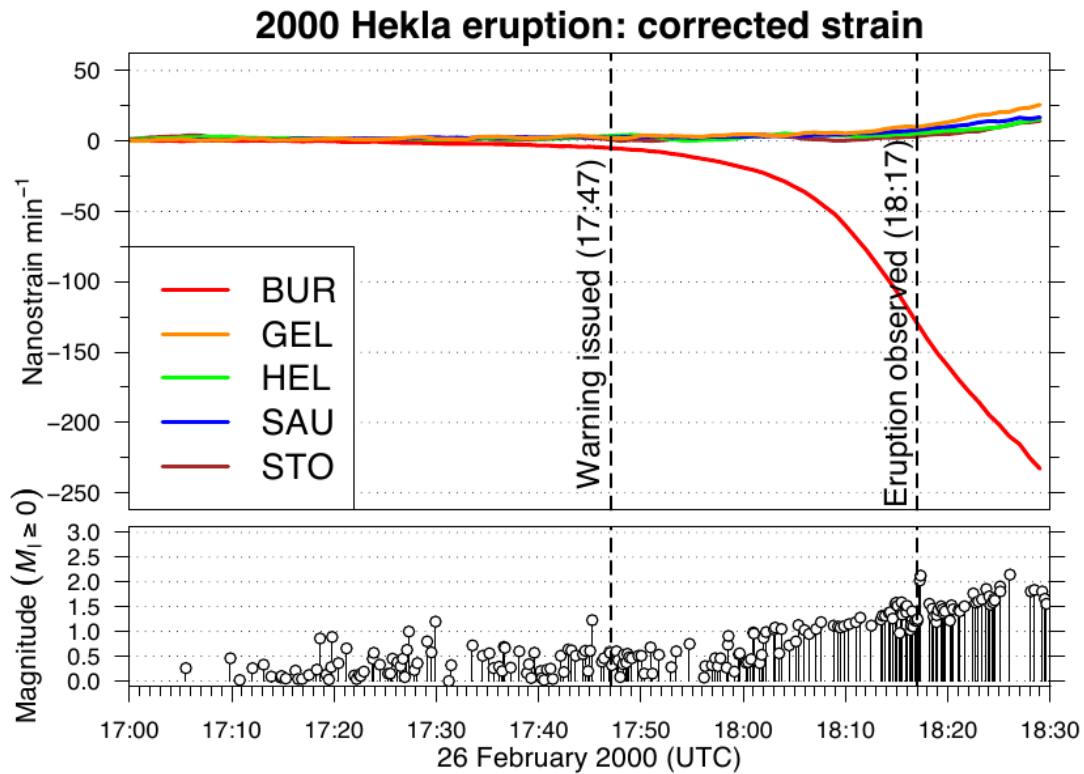


Figure 1: Volumetric strain-rate at five borehole strainmeters during the onset of the 2000 eruption of Hekla. Station BUR, located ~15 km north-west of the summit of Hekla, was sensitive to magma forcing its way from a reservoir at depth upwards to the surface (Linde et al., 1993). Note that expansion is positive. The lower plot shows micro-earthquakes detected at Hekla during the beginning of the eruption. Both data-sets are sourced from IMO. For station locations, see Figure 2.

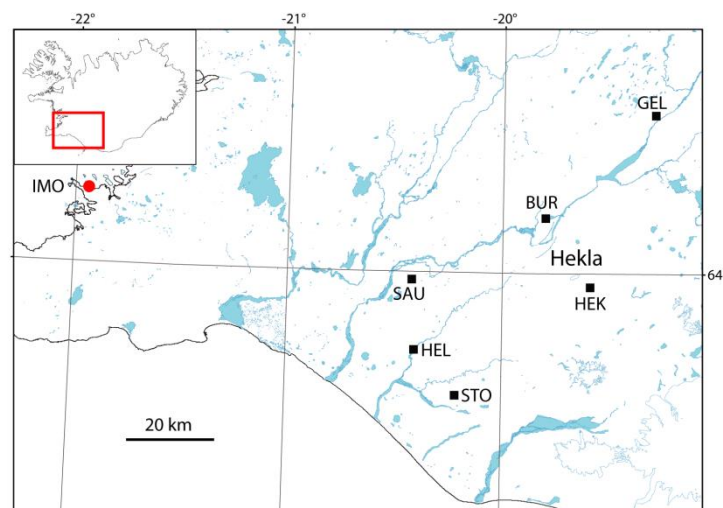


Figure 2: Schematic map of borehole strainmeters depicted in Figure 1. Note that HEK was installed in September 2010. In 2015, four stations were operational, as shown in Figure 3.

### Challenge of providing useable results

Developments in sensor design and data logging allow various types of geophysical data to be recorded continuously; collection rates of 1–10 samples per second (sps) are commonplace for survey-grade GPS receivers, whereas seismological sensors are logged typically at 20–200 sps (Guralp Systems Ltd, 2015; Trimble Navigation Ltd, 2015). For

borehole dilatometers such as the Sacks-Evertson single-component strainmeter, the sampling rate ranges 1–50 sps (Roeloffs and Linde, 2007). Likewise, sampling rates of 1 sps are routine for river-based hydrological sensors (Campbell Scientific, Inc., 2015). As data recording capabilities have improved, so too have modern telecommunication methods, particularly IP-based telemetry. High-rate data sampling and high-capacity telemetry techniques make data streaming viable. For various geophysical monitoring networks at IMO, the lag between actual data collection and delivery ranges from seconds (seismic and CGPS) to minutes (borehole strain and hydrological). Without dedicated handling of the incoming data, the result is masses of archived measurements but not enough useable results. Additionally, integrating diverse types of data into a format that can be used for monitoring purposes is a challenging task. Consequently, many organisations are unable to publish massive volumes of continuous data in near-real-time.

## Methods

The open-source R programming language was chosen to process and visualise data in this study. R is a free software environment for statistical computing and graphics (R Core Team, 2015). The software provides a wide variety of statistical and graphical techniques and it is highly extensible, making it ideal for handling diverse time-series. The software was in operational use already at IMO, where it has been used as (i) an effective data handling and storage environment; (ii) a facility for performing complex calculations on data arrays and matrices; and (iii) a means for graphically displaying almost any type of temporal or spatial data. These qualities made it a logical and convenient choice for near-real time processing and display of multi-parameter volcanological data. In the following examples, R version 3.0.1 (2013-05-16) and R version 3.1.1 (2014-07-10) were used. Commands were compiled into single, executable scripts that were interoperable between R versions. Automation was achieved by running R in 'slave mode' as a cron-job on a Debian GNU/Linux virtual server. Volumetric strain data were streamed to IMO as waveform files encoded in Seismic Analysis Code (SAC) binary format (IRIS Data Services, 2015). R package RSEIS (Lees, 2015) was used to return time and data vectors in alphanumeric format whilst maintaining native header formatting and file structure. Fully commented copies of the R scripts outlined next are enclosed as appendices.

## Hekla strain plot

Figure 1 emphasises the importance of continuous monitoring of Hekla. With this in mind, an R script was written to download 1 sps strain data at three minutes intervals from four volumetric strainmeters (BUR, HEK, HEL, and STO). Initial work on the Hekla strain plot was marked by Milestone MS47, accomplished in October 2013. In completed form, the script retrieves strain measurements in SAC format and then merges them into a single file. Strain data are written to file incrementally every second, with files spanning 24-hour periods from midnight to midnight, resulting in a maximum file-size of ~340 KB. Within the R script, the *rsync* Linux utility is used to synchronise files between site and data centre using SSH for additional security. The script was designed to allow data to be downloaded simultaneously from all stations, thus minimising time delays. The strain time-series is filtered using a 121-s median average; this removes short-period signals such as those caused by regional earthquakes, enabling easier recognition and automatic detection of deformation signals from Hekla.

The R script responsible for producing the Hekla strain plot operates at five-minute intervals. The plot combines near-real-time observations of borehole volumetric strain, seismic amplitude (tremor), and micro-earthquake activity to produce a live overview of the Hekla volcano (Figure 3).

Seismic data incorporated into the plot include filtered, one-minute-averaged tremor measurements (Böðvarsson *et al.*, 1999) from Haukadalur – an IMO seismic

station located 15 km west-south-west of Hekla. Only the horizontal components (N and S) are considered and the filtering is done in three frequency bands: 0.5–1 Hz, 1–2 Hz, and 2–4 Hz (Figure 3). The approach is similar to the Real-time Seismic-Amplitude Measurement (RSAM) developed by the USGS (Murray and Endo, 1989). Similar to RSAM, the tremor component of the Hekla plot provides a simplified yet useful measure of the overall level of seismic activity. The time-series is at its most valuable when the volcanic signal dwarfs other environmental noises – precisely the situation preceding most volcanic eruptions.

Earthquakes occur infrequently at Hekla – the volcano exhibits hardly any seismicity in between volcanic eruptions. Between 2001 and 2014, the mean, six-monthly earthquake-rate was  $\sim 2.5$  (IMO, unpublished data and the Catalogue of Icelandic volcanoes, D4.2), making any seismicity at the volcano noteworthy. During each plotting run, the R script searches for seismicity in the Hekla region from the SIL catalogue of automated earthquake detections, made by IMO's national seismic network. The co-ordinates of the search region are: 19.55–19.85° W by 63.93–64.07° N. The lower region of the plot has provision on the left-hand y-axis for earthquake magnitudes. When no seismicity is detected, the default magnitude range is  $M_l$  0 to 3 (Figure 3). The plot also features an earthquake counter in the lower right of the plotting region, which displays the cumulative number of earthquakes detected at Hekla (Figure 3).

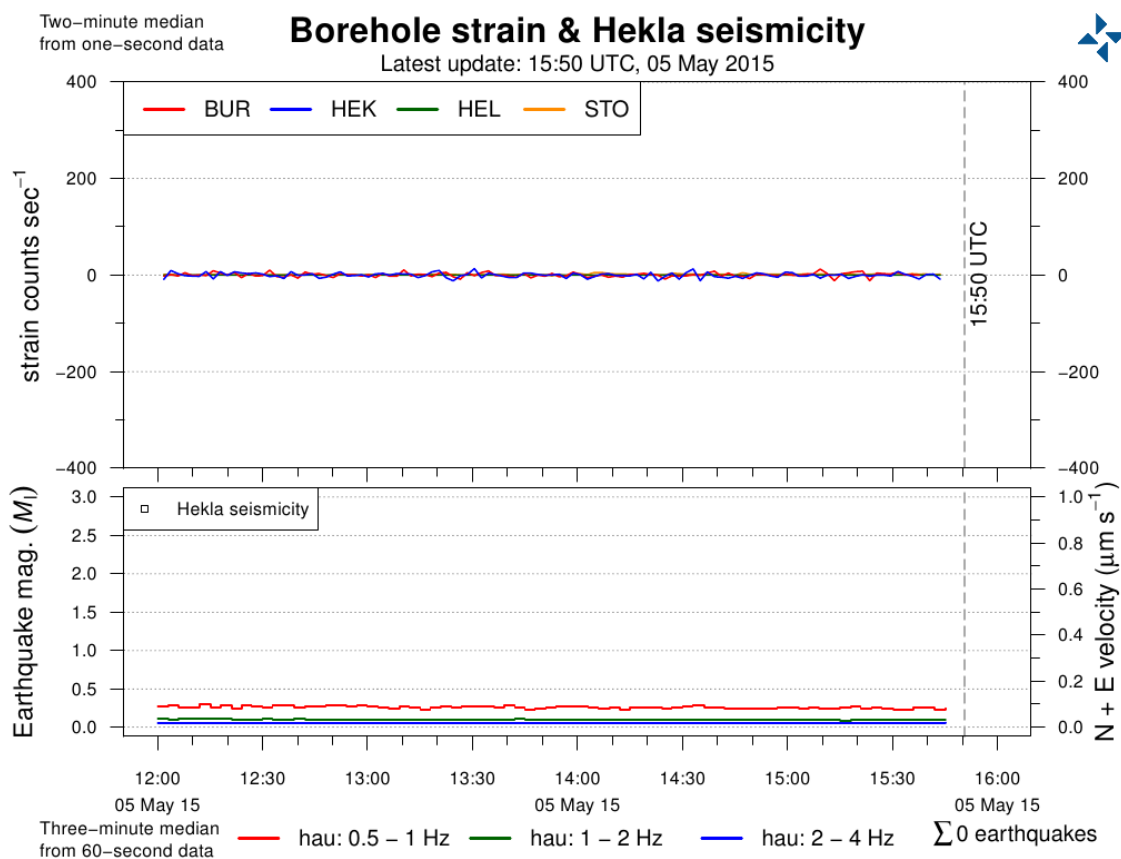


Figure 3: The Hekla strain plot. The upper plot displays the volumetric strain-rate (counts per second) at four borehole strainmeters. For station locations, see Figure 2. The lower plot shows filtered seismic amplitude measurements from Haukadalur. Note that the plot combines north and east seismic components to produce an overall measure of horizontal ground velocity. The left-hand axis of the lower plot is reserved for discrete earthquake activity at Hekla.

A PNG version of the vector-based plot is copied using SSH from the processing server to the web server (Figure 4), where it has been available to the public since 2013 via [http://hraun.vedur.is/ja/hekla/borholu\\_thensla.html](http://hraun.vedur.is/ja/hekla/borholu_thensla.html). At IMO, the plot serves as a

central monitoring tool for Hekla, although it is of wide interest to the public as well. Individual plots of 1 sps strain data are also linked to the Hekla plot; these plots show 48 hours of unprocessed data (see: <http://hraun.vedur.is/ja/strain/1sec/index.html>). The download, processing, and plotting of data has been designed for maximum reliability, allowing the webpage to be used by IMO for geophysical monitoring. Various enhancements have maximised the operational dependability of the Hekla strain plot. A series of stress tests at IMO on the hardware, data links, and processing routines behind the plot have helped to identify several potential failures. All known vulnerabilities have been addressed, resulting in a highly stable R environment. In fact, the plot will continue to function even if all strain data are missing. Figure 4 gives a structural overview of the routines behind the Hekla plot. Figure 5 illustrates one of several operational checks in place on the Hekla plot. A dedicated R script compares the creation time of the image version of the plot between processing and web servers; if the time difference exceeds ten minutes, IMO monitoring staff receive an e-mail message. Separate resource checks are also made on the processing server (e.g. CPU load and network services).

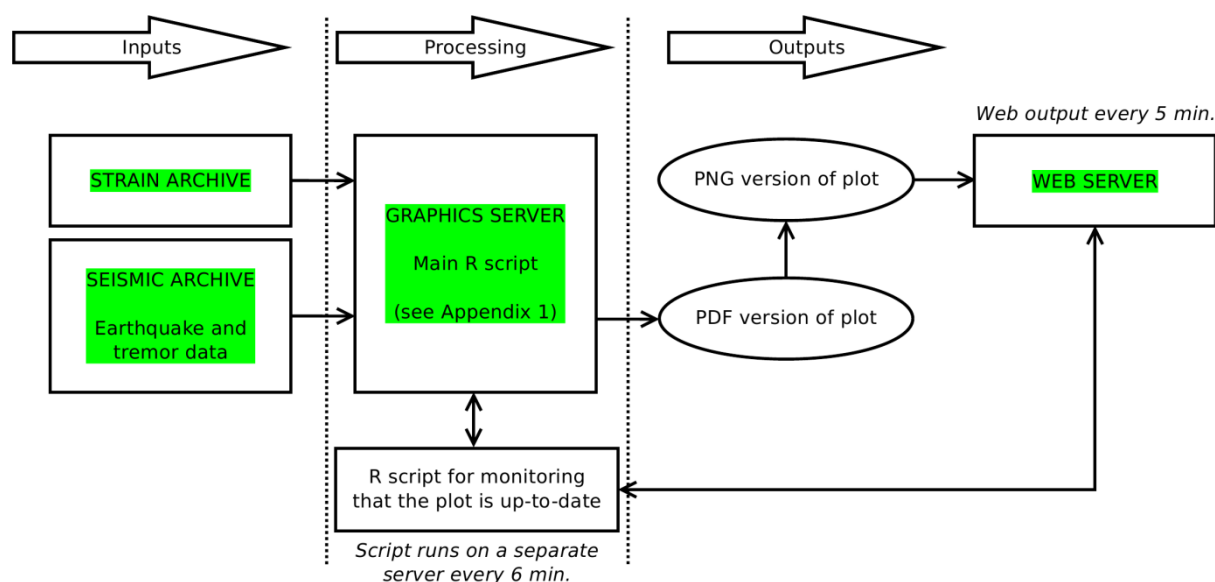


Figure 4: Flowchart showing the operation of the Hekla plot. The main R script (Appendix 1) sources the input data, producing PDF and PNG versions of the resulting plot (Figure 3). The latest PNG version of the plot is then copied to a separate web server: [http://hraun.vedur.is/ja/hekla/borholu\\_thensla.html](http://hraun.vedur.is/ja/hekla/borholu_thensla.html). Note the dedicated R script for checking that the plot is up-to-date (Figure 5).

```

----- Forwarded message -----
From: The STRAIN system user <[...].vedur.is>
Date: 8 May 2015 at 11:48
Subject: ATH: Hekla plot - time difference is too great
To: [...].vedur.is, [...].vedur.is
Cc: [...].vedur.is

This is an automated message, originating from [...].vedur.is

COMPARISON OF IMAGE CREATION TIMES

==> [...].vedur.is <==
    Date and Time      File path
    2015-05-08 11:45:17 /[...] /hekla_plot_20150508_114516.png

==> [...].vedur.is <==
    Date and Time      File path
    2015-05-08 11:25:16 /[...] /hekla_plot.png

Time difference of 20 mins

See: http://hraun.vedur.is/ja/hekla/borholu\_thensla.html

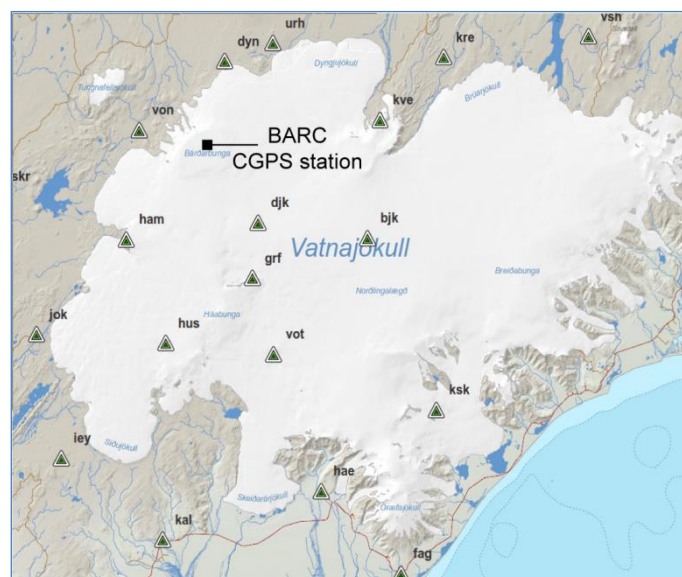
Contact: Eftirlitsmaður a jadvakt / Matthew J. Roberts

```

*Figure 5: Example of the R-based checks in place to ensure that the Hekla plot is up-to-date. Note that, for security reasons, sensitive network details have been removed, as denoted by "[...]". The plot's creation time is compared between processing and web servers (Figure 4). If the plot on the web server has not been updated within 10 minutes, an automated e-mail message is sent to IMO monitoring staff.*

### Bárðarbunga GPS plot

In mid-August 2014, IMO's volcanic monitoring focussed on seismic unrest and caldera subsidence at Bárðarbunga, an ice-capped volcano beneath north-west Vatnajökull. Following the onset of an intense seismic swarm at the volcano, airborne and satellite-based surveys revealed large-scale subsidence of the ice surface, signifying collapse of the caldera floor (Gudmundsson *et al.*, 2015). A CGPS station was installed on the ice surface of the caldera in early September to enable continuous elevation measurements (Figure 6). Positioning data were streamed at 1 sps via radio modems to a nearby SIL seismic station (Kverkjöll), and then via the Internet to IMO using 3G telecommunication (Figure 6).



*Figure 6: Map of Vatnajökull showing the location of SIL seismic stations and the Bárðarbunga CGPS station. Note the location of SIL station kve (Kverkjöll), which was used to relay geodetic data from the caldera of Bárðarbunga.*

The CGPS data were processed in real-time using RTKLIB – an open-source program for GNSS positioning (RTKLIB, 2015). Using R-based methods derived from the Hekla strain plot, an innovative, near-real-time display of caldera subsidence and seismicity was developed. Additional proficiency in CGPS data handling and visualisation came from a pilot project by IMO in 2012 on Mýrdalsjökull – the ice-cap overlying the Katla volcano. This project demonstrated the feasibility of streaming CGPS observations for monitoring of natural hazards (Figure 7). Graphical elements of both the Hekla strain plot and the Mýrdalsjökull CGPS plot were combined to form the basis of an R script for near-real-time plotting of caldera subsidence and seismicity (Appendix 2).

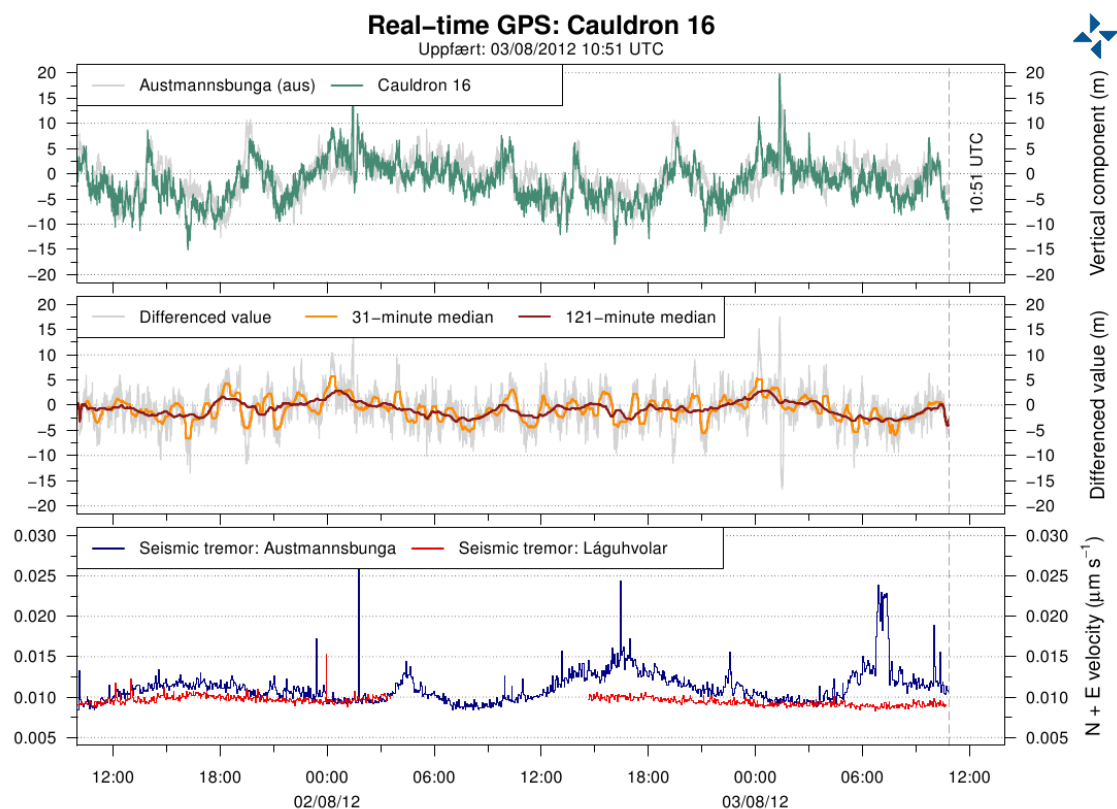


Figure 7: Experimental plot of CGPS data and seismic tremor measurements. The CGPS receiver was located in an ice cauldron on Mýrdalsjökull. The cauldron formed suddenly in July 2011 due to volcanic unrest in the Katla caldera, resulting in an outburst flood that caused damage to transport and communication infrastructure. The project further demonstrated how integrated observations could be used for monitoring purposes; it was effectively the forerunner to the Bárðarbunga CGPS plot (Figure 8).

Two R scripts were written to display geodetic and seismic data from Bárðarbunga. The first script was designed to handle the entire CGPS time-series from 11 September 2014 onwards (Figure 8). The script was adapted to tolerate a daily increase of 12 MB of CGPS data. By mid-February 2015, incremental archiving of CGPS data resulted in scripting runs with almost 1 GB of files; this confirmed that R is well-suited to handle large data-sets. The second script produced a running, 72-hour time-series of vertical displacement and corresponding seismicity in the caldera (Figure 9). The plot covering the entire time-series (Figure 8) was updated every 33 minutes, whereas the shorter, 72-hour plot (Figure 9) was created every five minutes. As soon as both plots were in operational use they were made available to the public through IMO webpages: <http://www.vedur.is/skjalftar-og-eldgos/gps-maelingar/bardarbunga/askjan/> and <http://hraun.vedur.is/ja/Bardarb/BARC/>.

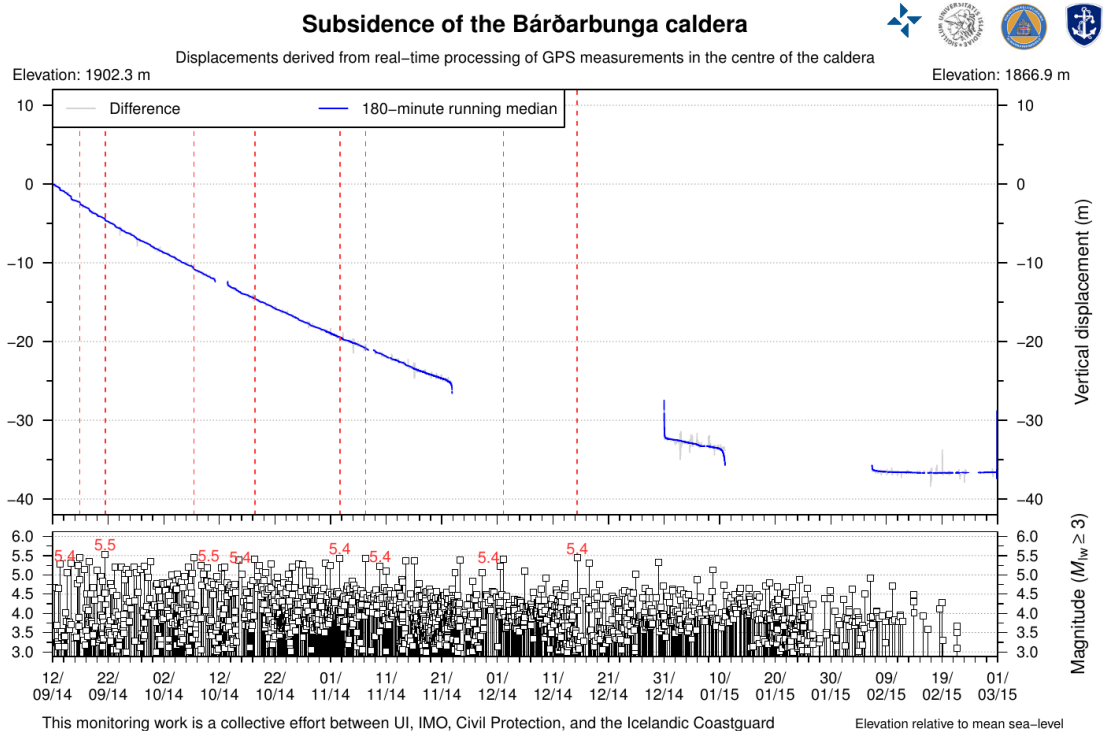


Figure 8: Online version of the Bárðarbunga CGPS plot, released publicly on 11 September 2014. The blue line denotes cumulative vertical displacement of the station in metres. Gaps in the time-series were due to data-transfer problems. The vertical, dashed lines signify caldera earthquakes with a moment-magnitude ( $M_w$ ) equal to or greater than five. (Over 79 earthquakes were depicted in this manner during the unrest period.) The lower region of the plot shows discrete earthquakes equal to or greater than  $M_w$  3.

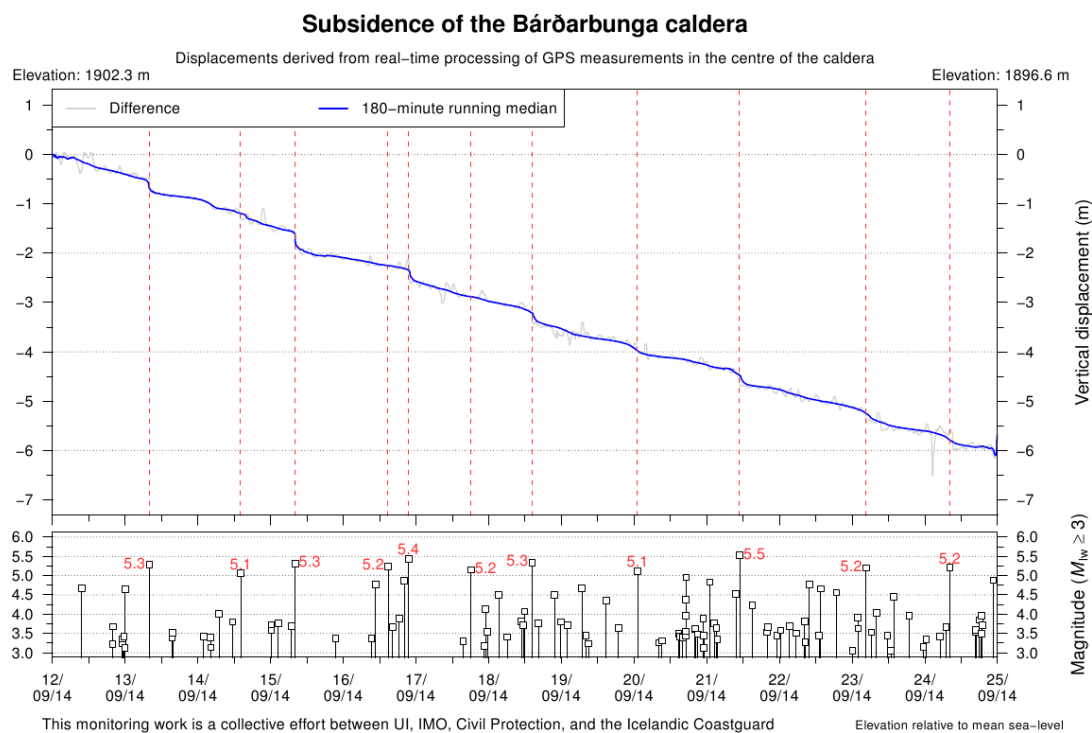


Figure 9: Example of the temporal correlation between larger caldera earthquakes and sudden lowering of the ice-surface, resulting in step-like displacements. For further details, see Figure 8.

## Conclusions

This study demonstrates how advances in high-rate data sampling and rapid data transfer can be exploited in an automated fashion for near-real-time processing and visualisation of monitoring results. By minimising the lag between time-of-measurement, receipt of monitoring data, and the availability of monitoring products, timely results can be used to make continuous assessments of eruption potential. The data handling approach outlined in this study emphasises the importance of visualising multi-parameter measurements in near-real-time. Such data visualisation and interoperability issues are a common problem in many scientific fields. As volcano observatories continue to address the challenge of integrating multi-parameter data in near-real-time, the R computing environment has proven to be well-suited for accessing and visualising diverse temporal data-sets. Open-source software such as R allows limitless data access, analysis, and visualisation possibilities. Similar development work should be prioritised at volcano observatories worldwide, as timely monitoring and processing results are the only scientific basis for short-range eruption early warnings (Tilling, 2008).

## Acknowledgements

I thank the following IMO staff for their assistance: Kristín S. Vogfjörð for comments on earlier drafts of this report; Benedikt G. Ófeigsson for collaboration on CGPS data streaming; and Einar Indriðason and Vigfús Gíslason for computing advice and operational support.

## References

Agustsson, K., R. Stefansson, A.T. Linde, P. Einarsson, I.S. Sacks, G.B. Gudmundsson, and B. Thorbjarnardottir, 2000, Successful prediction and warning of the 2000 eruption of Hekla based on seismicity and strain changes. *Eos Transactions AGU*, **81**, Fall meeting supplement, abstract V11B-30.

Böðvarsson, R., S.Th. Rögnvaldsson, R. Slunga, and E. Kjartansson, 1999, The SIL data acquisition system — at present and beyond year 2000. *Physics of the Earth and Planetary Interiors*, **113**, DOI: 10.1016/S0031-9201(99)00032-1.

Campbell Scientific, Inc., 2015, URL: <https://www.campbellsci.com/water-level-flow> (accessed: 2015-05-04).

Dzurisin, D., 2003, A comprehensive approach to monitoring volcano deformation as a window through the eruption cycle. *Reviews of Geophysics*, **41**, 1, DOI: 10.1029/2001RG000107.

Gudmundsson, M.T., K. Jónsdóttir, M. Roberts, B.G. Ófeigsson, Th. Högnadóttir, E. Magnússon, A.H. Jarosch, F. Pálsson, P. Einarsson, F. Sigmundsson, V. Drouin, V. Hjörleifsdóttir, H.I. Reynolds, T. Dürig, K. Vogfjörð, M. Hensch, J. Munoz-Cobo Belart, and B. Oddsson, 2015, The 2014–2015 slow collapse of the Bárðarbunga caldera, Iceland. *Geophysical Research Abstracts*, **17**, EGU General Assembly 2015, EGU2015-12521.

Guralp Systems Ltd, 2015, URL: <http://www.guralp.com/applications/seismology/volcano-monitoring> (accessed: 2015-05-04).

IRIS Data Services, 2015, URL: <http://www.iris.edu/hq/> (accessed: 2015-05-04).

- Lees, J.M., 2015, *RSEIS: Seismic time series analysis tools*. R package. URL: <http://cran.r-project.org/web/packages/RSEIS/> (accessed: 2015-05-04).
- Linde, A.T., K. Agustsson, I.S. Sacks, and R. Stefansson, 1993, Mechanism of the 1991 eruption of Hekla from continuous borehole strain monitoring. *Nature*, **365**, 6448, DOI: 10.1038/365737a0.
- Murray, T.L. and E.T. Endo, 1989, *A real-time seismic amplitude measurement system (RSAM)*. USGS open-file report, 89-684, 26 p.
- R Core Team, 2013. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org/> (accessed: 2015-05-04).
- Roeloffs, E.A. and A.T. Linde, 2007, Borehole observations of continuous strain and fluid pressure. In: Dzurisin, D. (ed.), *Volcano Deformation*, Springer-Praxis, Chichester, 305–322.
- RTKLIB, 2015, RTKLIB: An open source program package for GNSS positioning. URL: <http://www.rtklib.com/> (accessed: 2015-05-04).
- Sigmundsson, F., A. Hooper, S. Hreinsdottir, K.S. Vogfjord, B.G. Ofeigsson, E.R. Heimisson, S. Dumont, M. Parks, K. Spaans, G.B. Gudmundsson, V. Drouin, T. Arnadottir, K. Jonsdottir, M.T. Gudmundsson, T. Hognadottir, H.M. Fridriksdottir, M. Hensch, P. Einarsson, E. Magnusson, S. Samsonov, B. Brandsdottir, R.S. White, T. Agustsdottir, T. Greenfield, R.G. Green, A.R. Hjartardottir, R. Pedersen, R.A. Bennett, H. Geirsson, P.C. La Femina, H. Bjornsson, F. Palsson, E. Sturkell, C.J. Bean, M. Mollhoff, A.K. Braidon, and E.P.S. Eibl, 2015, Segmented lateral dyke growth in a rifting event at Bardarbunga volcanic system, Iceland, *Nature*, **517**, 7533, DOI: 10.1038/nature14111.
- Sigmundsson, F., S. Hreinsdottir, A. Hooper, T. Arnadottir, R. Pedersen, M.J. Roberts, N. Oskarsson, A. Auriac, J. Deciem, P. Einarsson, H. Geirsson, M. Hensch, B.G. Ofeigsson, E. Sturkell, H. Sveinbjornsson, and K.L. Feigl, 2010, Intrusion triggering of the 2010 Eyjafjallajokull explosive eruption, *Nature*, **468**, 7322, DOI: 10.1038/nature09558.
- Tilling, R.I., 1989, Volcanic hazards and their mitigation - progress and problems. *Reviews of Geophysics*, **27**, 2, DOI: 10.1029/RG027i002p00237.
- Tilling, R.I., 2008, The critical role of volcano monitoring in risk reduction. *Advances in Geosciences*, **14**, DOI: 10.5194/adgeo-14-3-2008.
- Trimble Navigation Ltd, 2015. URL: <http://www.trimble.com/Infrastructure/Earth-Systems.aspx> (accessed: 2015-05-04).

## Appendix 1: R script for the Hekla strain plot

*N.B. Sensitive network details have been removed, as denoted by "[...]"*

```
#Matthew J. Roberts: matthew@vedur.is
#05/05/2015
#R: A language and environment for statistical computing
#R version 3.1.1 (2014-07-10)

options(digits.secs=3) #Digits to print when formatting digital seconds

setwd("/[...]/")

t1 <- as.POSIXlt(cut.POSIXt(Sys.time(),"day"))-86400 #System time minus 24 hours (in seconds)
t2 <- t1+86400 #'t1' plus 24 hours (in seconds)

#For compatibility with POSIX time-stamps in the data.frame
t1 <- as.POSIXct(t1)
t2 <- as.POSIXct(t2)

aa <- data.frame(TIME=seq.POSIXt(t1,t2,"1 day"))
aa <- format(aa,"%Y/%b/%d")

### Read earthquake data ###
bb1 <- tolower(paste("/[...]/",paste(aa[,1],"/",sep=""),"aut.mag",sep=""))
#Wrapper 'try' used to trap and suppress errors due to missing files
cc1 <-lapply(bb1,function(x){
  try(
    na.omit(read.table(x,header=F,sep=" ",
      colClasses=c("numeric",rep("character",2),rep("numeric",5),"character"))),silent=T)
})

#Omit class 'try.error' by logical test for 'data.frames'
cc1 <- cc1[which(sapply(cc1,class)=="data.frame")]

dd <- do.call(rbind,cc1); rm(cc1)

dd[,1] <- as.POSIXct(strptime(paste(substring(dd[,2],1,8),
  substring(dd[,3],1,8)),format="%Y%m%d %H%M%S"))

dd <- data.frame(dd[,c(2:3,8:9)]) #Remove unnecessary variables

#Test for the case of an 'empty' aut.mag file (e.g. when the script runs just after midnight)
if(nrow(dd)==0) dd <- as.data.frame(rep(list(a=numeric(0)),5))

names(dd) <- c("TIME","LATITUDE","LONGITUDE","DEPTH","ML")

#Sub-set data.frame based on Hekla alert region: /[...]/region.index
dd <- subset(dd,dd[,2]>=63.93 & dd[,2]<=64.07 & dd[,3]>=-19.85 & dd[,3]<=-19.55)

#Nested 'ifelse' tests to replace spurious magnitude values
#Usage: ifelse(test,yes,no)
#If the condition of the first argument is FALSE,
#then the third argument is evaluated (i.e. another 'ifelse')
dd[,5] <- ifelse(dd[,5]<=-9,1,ifelse(dd[,5]>=9,1,dd[,5]))

dd[,5] <- as.numeric(dd[,5]) #Revert Ml to class 'numeric'

### Read 'SAC' strain data ###
library(RSEIS)

#Wildcard selection of 1-second 'sla' files
ee <- Sys.glob(noquote(paste("/[...]/*.s?a.*.1s.sac",sep="")))

ee1 <- ee[file.info(ee)$size > 1] #Prevent 'empty' files from being read

ee2 <- ee[file.info(ee)$size < 1] #Files less than one byte in size

file.remove(list=(ee2)) #Remove 'empty' files (forces [...].vedur.is to download file again)

ee <- as.data.frame(ee1)

ee[,2] <- as.POSIXct(strptime(paste(substring(ee[,1],40,58)),format="%Y.%m.%d.%H.%M.%S"))

ee <- ee[order(ee[,2],decreasing=T),] #Rearrange POSIX times in descending order
```

```

ee[,3] <- substring(ee[,1],32,58) #Extract station name, parameter and time-stamp

ee <- ee[!duplicated(ee[,3]),] #Remove duplicated elements from the data.frame

ee <- as.character(ee[,1]) #Coerce data.frame to 'character' object

#Read SAC binary data using 'RSEIS' package
ff <- JSAC.seis(ee,Iendian=1,BIGLONG=F) #SAC inserts -12345 for no data; see: 'ff[[1]]$HEAD'

#Set object names in list 'ff'
names(ff) <- lapply(seq(length(ee)),function(x){names(ff)[x] <- substring(ee[x],32,58)})

gg <- list() #Empty lists, created outside the following loop
hh <- list()
ii <- list()

for (x in names(ff))
{
  jj <- as.POSIXct(strptime(paste(ff[[x]]$DATTIM$j,d,ff[[x]]$DATTIM$yr,
ff[[x]]$DATTIM$hr,ff[[x]]$DATTIM$mi,ff[[x]]$DATTIM$msec),"%j %Y %H %M %OS")) #See '?strptime'

  ff[[x]]$TIME <- seq.POSIXt(jj,jj+(ff[[x]]$N*ff[[x]]$dt-1),"sec") #Generate POSIX sequence

  gg[x] <- lapply(ff[x],function(x){c("TIME","amp")}) #Extract vectors from list

#Return vectors of selected elements
  hh[[x]] <- lapply(gg[x][1],function(x){c("TIME")})
  ii[[x]] <- lapply(gg[x][2],function(x){c("amp")})
}

#Index elements from list; see 'str(hh)'
#Ordering of station names cannot be assumed beforehand, hence names set using 'assign'
for (x in seq(length(names(hh))))
{
  assign(names(hh[x]),data.frame(hh[x][[1]],ii[x][[1]]))
}

#Substitute data.frame, in case SAC files are missing
tmp <- data.frame(TIME=seq(as.POSIXct(cut.POSIXt(Sys.time(),"hour"))-2*3600,
by="sec",length.out=2*3600),amp=NA)

bur <- as.list(mget(ls(pattern="bur"),envir=.GlobalEnv))
bur <- do.call(rbind,bur)
if(inherits(bur,"NULL")){bur <- tmp}

hek <- as.list(mget(ls(pattern="hek"),envir=.GlobalEnv))
hek <- do.call(rbind,hek)
if(inherits(hek,"NULL")){hek <- tmp}

hel <- as.list(mget(ls(pattern="hel"),envir=.GlobalEnv))
hel <- do.call(rbind,hel)
if(inherits(hel,"NULL")){hel <- tmp}

sto <- as.list(mget(ls(pattern="sto"),envir=.GlobalEnv))
sto <- do.call(rbind,sto)
if(inherits(sto,"NULL")){sto <- tmp}

#Coerce frames into one; 'NAs' included for padding
sac <- merge(bur,hek,by.x="TIME",by.y="TIME",all=T)
colnames(sac)[2:3] <- c("BUR","HEK")
sac <- merge(sac,hel,by.x="TIME",by.y="TIME",all=T)
colnames(sac)[4] <- "HEL"
sac <- merge(sac,sto,by.x="TIME",by.y="TIME",all=T)
colnames(sac)[5] <- "STO"

#Write data.frame to file
system("rm -f sac_ls_orig.dat")
write.table(cbind(sac[,1],sac[,2:5]),file="sac_ls_orig.dat",sep="\t",row.names=F,col.names=T)

#Lagged differences
sac[,2:5] <- data.frame(lapply(sac[,2:5],function(x){c(diff(x,differences=1),NA)}))
sac <- sac[-nrow(sac),] #Omit last row from data.frame

sac[,1] <- as.POSIXct(sac[,1],origin="1970/01/01")

### Filter strain data using a 121-s median average ###

```

```

sac1 <- data.frame(lapply(sac[,2:5],function(x){aggregate(as.numeric(x),
list(cut.POSIXt(sac[,1],"121 sec")),median)}))
sac1 <- sac1[-nrow(sac1),] #Omit last row from data.frame
sac1 <- sac1[,-c(3,5,7)]

names(sac1) <- c("TIME","BUR","HEK","HEL","STO")
sac1[,1] <- as.POSIXct(sac1[,1],origin="1970/01/01")

t1 <- as.POSIXct(cut.POSIXt(Sys.time(),"hour"))-3*3600 #Minus three hours
t2 <- as.POSIXct(cut.POSIXt(Sys.time(),"hour"))+3600 #Plus one hour

#Sub-set 'SAC' data.frame
sac2 <- list(sac,sac1) #Construct list
sac2 <- lapply(sac2,function(x){subset(x,x[,1]>=t1 & x[,1]<=t2)})

sac <- as.data.frame(sac2[1]) #Return data.frame
sac1 <- as.data.frame(sac2[2])

#Remove all objects aside from specified data.frames
rm(list=ls()[!(ls()%in%c("dd","sac","sac1","t1","t2"))])

#Sub-set data.frame to ensure that the earthquake counter only includes events on the graph
dd <- subset(dd,dd[,1]>=t1 & dd[,1]<=t2)

### Read tremor data ###
#Collate tremor data using 'gettrem'

gettrem <- "/[.]"

stn <- noquote("hau") #SIL seismic station

aal <- paste(gettrem," -b 2"," -c1 -s ",stn," > ",stn,"_n_05_1.dat;",sep="") #North: 0.5-1 Hz
bb1 <- paste(gettrem," -b 2"," -c2 -s ",stn," > ",stn,"_e_05_1.dat;",sep="") #East: 0.5-1 Hz
cc1 <- paste(gettrem," -b 2"," -c4 -s ",stn," > ",stn,"_n_1_2.dat;",sep="") #North: 1-2 Hz
dd1 <- paste(gettrem," -b 2"," -c5 -s ",stn," > ",stn,"_e_1_2.dat;",sep="") #East: 1-2 Hz
ee1 <- paste(gettrem," -b 2"," -c7 -s ",stn," > ",stn,"_n_2_4.dat;",sep="") #North: 2-4 Hz
ff1 <- paste(gettrem," -b 2"," -c8 -s ",stn," > ",stn,"_e_2_4.dat;",sep="") #East: 2-4 Hz

system(noquote(paste(aal,bb1,cc1,dd1,ee1,ff1)))

#Substitute data.frame, in case tremor data are missing
tmp <- data.frame(format(seq(c(ISOdate(1970,1,1)),by="min",length.out=60),"%Y%m%d%H%M%S"),1)

aal <- try(read.table(paste(stn,"_n_05_1.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(aal,"try-error")){aal <- tmp}

bb1 <- try(read.table(paste(stn,"_e_05_1.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(bb1,"try-error")){bb1 <- tmp}

cc1 <- try(read.table(paste(stn,"_n_1_2.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(cc1,"try-error")){cc1 <- tmp}

dd1 <- try(read.table(paste(stn,"_e_1_2.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(dd1,"try-error")){dd1 <- tmp}

ee1 <- try(read.table(paste(stn,"_n_2_4.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(ee1,"try-error")){ee1 <- tmp}

ff1 <- try(read.table(paste(stn,"_e_2_4.dat",sep=""),header=F,sep="",
colClasses=c("character","numeric")),silent=T)
if(inherits(ff1,"try-error")){ff1 <- tmp}

hau <- cbind(aal,bb1[,2],cc1[,2],dd1[,2],ee1[,2],ff1[,2])

hau[,1] <- as.POSIXct(strptime(paste(hau[,1]),format="%Y%m%d%H%M%S"))

hau[,8] <- round(sqrt(hau[,2]^2+hau[,3]^2),3) #Total displacement vector
hau[,9] <- round(sqrt(hau[,4]^2+hau[,5]^2),3)
hau[,10] <- round(sqrt(hau[,6]^2+hau[,7]^2),3)

hau <- hau[,-c(2:7)]

```

```

#Velocity at hau
#LE5 sensor: 400 V/m/s; G24 digitiser: 1/384615.38 micro-Volt/bit
#count * 2.6e-06 / 400 = count * 2.6e-06 = count * nanometre/second
#2.6e-6/400 * 1e06 = 0.0065 #micro-metres / second
hau[,2:4] <- cbind(hau[,2:4]*0.0065)

### Filter tremor data using a three-minute median average ###
trem <- data.frame(lapply(hau[,2:4],function(x){aggregate(as.numeric(x),
list(cut.POSIXt(hau[,1],"3 min")),median)}))
trem <- trem[,c(3,5)]

trem[,1] <- as.POSIXct(trem[,1])

names(trem) <- c("TIME","hau05_1","hau1_2","hau2_4")

#Sub-set tremor data.frame
hau <- subset(trem,trem[,1]>=t1 & trem[,1]<=t2)

#Remove all objects aside from specified data.frames
rm(list=ls()[!(ls() %in% c("dd","hau","sac","sac1","stn","t1","t2"))])

### Axes ###
#Formatting of Y-axis tick labels using 'pretty' function
#Values are chosen so that they are 1, 2 or 5 times a power of 10
sac[is.na(sac)] <- 0 #Replace 'NA' values with zeros

#lab1: Nested 'ifelse' tests to ensure a minimum x-axis range
lab1 <- pretty(ifelse(nrow(sac)==0,-100,ifelse(min(floor(sac[,2:4]))<=-
100,min(floor(sac[,2:4])),100)):ifelse(nrow(sac)==0,100,ifelse(max(ceiling(sac[,2:4]))>=100,
max(ceiling(sac[,2:4])),100)))
lab2 <- seq(min(lab1),max(lab1),diff(lab1)[1]/2)
lab3 <- pretty(ifelse(nrow(dd)==0,0,min(floor(dd[,5]))):ifelse(nrow(dd)==0,3,
max(ceiling(dd[,5])))) #Earthquakes
lab4 <- pretty(ifelse(nrow(hau)==0,0,min(floor(hau[,c(2:4)]))):ifelse(nrow(hau)==0,10,
max(ceiling(hau[,c(2:4)]))) #Seismic tremor
lab5 <- seq(min(lab4),max(lab4),diff(lab4)[1]/2)

### Legend variables ###
qu <- ifelse(nrow(dd)==1," earthquake"," earthquakes")
kk <- substitute(sum(paste(x,y)),list(x=sum(nrow(dd)),y=qu))
ll <- Sys.time()

### Set-up device ###
mm <- noquote(paste("hekla_plot_",format(ll,"%Y%m%d_%H%M%S"),".pdf",sep=""))

pdf(file=mm,title="matthew@vedur.is",
paper="special",width=8.5,height=6.5,bg="white",family="Helvetica")

par(oma=c(5,4.5,3,4)) #Outer margins
par(cex.axis=0.8) #Text size of labels
par(las=1)

### Uppermost graph: Strain-rate ###
par(mar=c(0,0,0,0)) #Plot margins (no. of lines). Base; left; top; right
par(fig=c(0,1,0.41,1)) #41-100% of height of fig. region: c(x1, x2, y1, y2)

plot.default(sac1[,1],sac1[,2],type="n",col="red",lwd=1.2,lty=1,las=1,xaxs="r",yaxs="i",
xlim=range(t1,t2),ylim=range(lab1),ann=F,axes=F,xaxt="n",panel.first=grid(NA,NULL,col="black",
lwd=0.8,lty=3)) #BUR

### Plot original, differenced data ###
lines(sac1[,1],sac1[,5],type="l",lwd=1.2,lty=1,col="darkorange") #STO
lines(sac1[,1],sac1[,4],type="l",lwd=1.2,lty=1,col="darkgreen") #HEL
lines(sac1[,1],sac1[,2],type="l",lwd=1.2,lty=1,col="red") #BUR
lines(sac1[,1],sac1[,3],type="l",lwd=1.2,lty=1,col="blue") #HEK
axis.POSIXct(1,range(t1,t2),at=seq(t1,t2,by="10 min"),"%H",labels=F,tcl=-0.3)
axis.POSIXct(1,range(t1,t2),at=seq(t1,t2,by="30 min"),"%H:%M",labels=F,tcl=-0.5,cex.axis=0.8)
par(las=1)
axis(2,at=lab1,labels=T,tcl=-0.5)
axis(2,at=lab2,labels=F,tcl=-0.3)
axis(4,at=lab1,labels=T,tcl=-0.5)
axis(4,at=lab2,labels=F,tcl=-0.3)
abline(v=ll,col="darkgrey",lwd=1.2,lty=5)
text(as.numeric(unclass(ll)+250),lab1[ceiling(length(lab1)/2)],
labels=paste(format(ll,"%H:%M"),"UTC"),cex=1,srt=90)

```

```

par(las=0)
mtext(expression(paste("strain counts sec"-1, sep="")), side=2, line=3, cex=1.2)
legend("topleft", legend=c("BUR", "HEK", "HEL", "STO"), ncol=4,
col=c("red", "blue", "darkgreen", "darkorange"), lwd=2, lty=1, bg="white", box.lty=1, inset=0, cex=1)

box()

### Lowermost graph: Seismic tremor and earthquakes ###
par(mar=c(0,0,0,0)) #Plot margins (no. of lines). Base; left; top; right
par(fig=c(0,1,0,0.38), new=T) #38-100% of height of fig. region: c(x1, x2, y1, y2)

plot(hau[,1], hau[,2], type="s", col="red", lwd=1.2, lty=1, las=1, xaxs="r",
yaxs="r", xlim=range(t1,t2), ylim=range(lab4), ann=F, axes=F, xaxt="n") #05-1 Hz
lines(hau[,1], hau[,3], type="s", col="darkgreen", lwd=1.2, lty=1) #1-2 Hz
lines(hau[,1], hau[,4], type="s", col="blue", lwd=1.2, lty=1) #2-4 Hz
par(las=1)
axis(4, at=lab5, labels=F, tcl=-0.3)
axis(4, at=lab4, labels=T, tcl=-0.5)
par(new=T)
plot.default(dd[,1], dd[,5], type="h", col="black", lwd=1, lty=1, las=1, xaxs="r", yaxs="r", xlim=range(
t1,t2),
ylim=range(lab3), ann=F, axes=F, xaxt="n", panel.first=grid(NA, NULL, col="black", lwd=0.8, lty=3))
#Earthquakes
points(dd[,1], dd[,5], pch=22, lwd=1, lty=1, col="black", bg="white", cex=1) #Overlay data points
axis.POSIXct(1, range(t1,t2), at=seq(t1,t2, by="10 min"), "%H", labels=F, tcl=-0.3)
axis.POSIXct(1, range(t1,t2), at=seq(t1,t2, by="30 min"), "%H:%M", labels=T, tcl=-0.5, cex.axis=0.8)
axis.POSIXct(1, range(t1,t2), at=seq(t1,t2, by="2 hour"), "%d %b %y",
labels=T, tcl=-0.5, line=1, lty=0)
axis(2, at=lab3, labels=T, tcl=-0.5)
abline(v=l1, col="darkgrey", lwd=1.2, lty=5)
par(las=0)
mtext(quote(Earthquake~mag.~(italic(M)[1])), side=2, line=3, cex=1.2)
mtext(expression(paste("N + E velocity (", italic(mu), "m s"-1, ")")), side=4, line=2.5, cex=1.2)
legend("topleft", legend="Hekla seismicity", pch=22, lwd=1, lty=0,
pt.bg="white", bg="white", box.lty=1, cex=0.8)
box()

title("Borehole strain & Hekla seismicity", line=1.5, outer=T, cex.main=1.6)
mtext(paste("Latest update:", format(l1, "%H:%M UTC, %d %B
%Y")), side=3, line=0.2, outer=T, cex.sub=0.8)

par(oma=c(1,1,1,1), new=T) #Fake outer margins
print(par("usr")) #A vector of the form 'c(x1, x2, y1, y2)'
plot.new() #Advance to a new graphics frame
par(xpd=NA) #If 'NA', all plotting is clipped to the device region
legend("bottom", legend=c(paste(stn, ": 0.5 - 1 Hz", sep=""), paste(stn, ": 1 - 2 Hz", sep=""),
paste(stn, ": 2 - 4 Hz", sep="")), inset=-0.07, ncol=3, xjust=0.5, yjust=0.5, lwd=2, lty=1,
col=c("red", "darkgreen", "blue"), bg="transparent", box.lty=0, cex=1)
mtext("Three-minute median\nfrom 60-second data", side=1, adj=0.01, outer=T, line=-0.3, cex=0.8)
mtext("Two-minute median\nfrom one-second data", side=3, adj=0.01, outer=T, line=-1, cex=0.8)
mtext(kk, side=1, adj=0.985, outer=T, line=-0.6, cex=1) #Earthquake counter

dev.off()

nn <- noquote(paste(gsub("\\.pdf", "", mm), ".png", sep=""))

system(paste("pdftoppm -r 120 -png", mm, ">", nn, sep=" ")) #Faster conversion, sharper graphics

system(paste("composite -compose atop -gravity NorthEast -geometry +10+10
/[...]/imo_logo.gif", nn, nn, sep=" "))

#Copy the latest PNG image to the web server
system(paste("scp", nn, "[...]"))

#Rename copied PNG image on the web server
system(paste("ssh [...]", shQuote(paste("cp ", "[...]/", nn, " /[...]/hekla_plot.png", sep=""))))
#Remove PNG images and PDF files older than 5*24 hours on the graphics server
system("find /[...]/hekla_plot*?? -mtime +5 -exec rm '{}' ';'")
#Remove PNG images older than 5*24 hours from web server
system(paste("ssh [...]", shQuote("find /[...]/hekla_plot*.png -mtime +5 -exec rm '{}' ';'")))

rm(list=ls()) #Remove objects from working environment
quit("no")
#EOF

```

## Appendix 2: R script for the Bárðarbunga CGPS plot

N.B. Sensitive network details have been removed, as denoted by "[...]".

```
#Matthew J. Roberts: matthew@vedur.is
#05/05/2015
#R: A language and environment for statistical computing
#R version 3.1.1 (2014-07-10)

setwd("/[...]/") #[...].vedur.is

t1 <- as.POSIXct("2014-09-12 00:00:00",origin="1970-01-01") #Define start date
t2 <- as.POSIXct(cut.POSIXt(Sys.time(),"hour"))

aa <- seq.POSIXt(t1,t2,"1 day")
aa <- format(aa,paste("barc","%Y%m%d",".dat",sep=""))

print(as.data.frame(aa))

t2 <- as.POSIXct(cut.POSIXt(Sys.time(),"hour"))+3600*144 #Plus six days
t3 <- as.POSIXct(format.POSIXt(t1,"%Y-%m-01")) #Truncate to beginning of month
t4 <- as.POSIXct(format.POSIXt(t2,"%Y-%m-01"))

#Read CGPS files
#Wrapper 'try' used to trap and suppress errors due to missing files
bb <- lapply(aa,function(x){
  try(
    read.table(x,header=F,sep=" ",na.strings="",fill=T,
      colClasses=c(rep("character",2),rep("numeric",13)),silent=T))
})

#Omit class 'try.error' by logical test for 'data.frames'
bb <- bb[which(sapply(bb,class)=="data.frame")]

cc <- do.call(rbind,bb); rm(bb)

cc[,1] <- as.POSIXct(strptime(paste(cc[,1],cc[,2]),format="%Y/%m/%d %H:%M:%OS"))

cc <- data.frame(cc[,-2]) #Remove unnecessary variable

cc <- cc[order(cc[,1],decreasing=F),] #Rearrange POSIX times in ascending order

cc <- data.frame(subset(cc,!duplicated(cc[,1]))) #Remove duplicate values, maintaining data

cc <- subset(cc,cc[,1]>=t1 & cc[,1]<=t2) #Sub-set data period

dd <- subset(cc,cc[,5]!=6) #Remove low-quality measurements

rm(cc)
gc() #Garbage collection - return some memory to OS

dd <- dd[,-c(2:3,6:14)] #Drop redundant columns

dd <- aggregate(dd[,2],list(cut.POSIXt(dd[,1],"1 min")),median) #One-minute median

dd[,1] <- as.POSIXt(dd[,1]) #Assign class POSIXt

colnames(dd) <- c("TIMESTAMP","ELEVATION") #Assign column names

#Sub-set data due to repositioning of station
t5 <- as.POSIXct("2014-11-11 12:00:00",origin="1970-01-01") #Repositioning work begins
t6 <- as.POSIXct("2014-11-11 19:00:00",origin="1970-01-01") #In place and stable

dd <- subset(dd,!dd[,1] %in% seq.POSIXt(t5,t6,by="min")) #! operator, logical

#Off-set correction following repositioning on 11/11/2014
dd[,2] <- ifelse(dd[,1]>=t5,dd[,2]-1.3,dd[,2]) #ifelse(test,yes,no)

#Off-set correction following repositioning on 05/02/2015
dd[,2] <- ifelse(dd[,1]>=as.POSIXt("2015-02-05"),dd[,2]-4.5,dd[,2]) #ifelse(test,yes,no)

rownames(dd) <- NULL #Remove row names

dd[,2] <- dd[,2]-67.71468 #Correction for height above mean sea level

dd[,3] <- cumsum(data.frame(c(0,diff(dd[,2],differences=1)))) #Differenced values
```

```

dd[,4] <- as.numeric(runmed(dd[,3],61,endrule="median")) #One-minute running median
dd[,5] <- cumsum(data.frame(c(0,diff(dd[,4],differences=1)))) #Median differenced values
dd[,6] <- as.numeric(runmed(dd[,3],10801,endrule="median")) #180-minute running median
dd[,7] <- cumsum(data.frame(c(0,diff(dd[,6],differences=1)))) #Median differenced values

colnames(dd)[3:7] <- c("DIFF","RUNMED1","RUNMED_DIFF1","RUNMED2","RUNMED_DIFF2")

#For 'padding' purposes
ddl <- data.frame(TIMESTAMP=seq.POSIXt(min(range(dd[,1])),max(range(dd[,1])), "1 min"),
ELEVATION=NA,DIFF=NA,RUNMED1=NA,RUNMED_DIFF1=NA,RUNMED2=NA,RUNMED_DIFF2=NA)

dd <- rbind(dd,ddl) #Combine actual and 'padded' data by rows

dd <- dd[order(dd[,1]),] #Ascending time-stamps

#Remove duplicate values, maintaining actual data
dd <- subset(dd,!duplicated(dd[,1]))

Sys.sleep(5)

#Read earthquake data (lib.mag)
ee <- data.frame(TIMESTAMP=seq.POSIXt(t1,Sys.time(),"1 day"))
ee <- format(ee,"%Y/%b/%d")

ff <- tolower(paste("/[.]/",paste(ee[,1],"/",sep=""),"lib.mag",sep=""))
#Wrapper 'try' used to trap and suppress errors due to missing files
gg <-lapply(ff,function(x){
  try(
    na.omit(read.table(x,header=F,sep="",
colClasses=c("numeric",rep("character",2),rep("numeric",20),"character"),
fill=T,na.string="*****")),silent=T)
})

#Omit class 'try.error' by logical test for 'data.frames'
gg <- gg[which(sapply(gg,class)=="data.frame")]

hh <- do.call(rbind,gg); rm(gg)

hh[,24][hh[,24]==""] <- "qu" #Assign filled entries as type 'qu'

hh[,1] <- as.POSIXct(strptime(paste(substring(hh[,2],1,8),
substring(hh[,3],1,10)),format="%Y%m%d %H%M%OS"))

hh <- hh[,-c(2:3,9:24)] #Drop redundant columns

colnames(hh) <- c("TIMESTAMP","LAT","LON","DEPTH","M_MAG","L_MAG")

#Define new function for sub-setting earthquakes in regions
area <- function(lat1,lat2,lon1,lon2){subset(hh,hh[,2]>=lat1 &
hh[,2]<=lat2 & hh[,3]>=lon1 & hh[,3]<=lon2)}

hh <- area(64.5885,64.7016,-17.6510,-17.3300) #Bárðarbunga caldera

rownames(hh) <- NULL #Remove row names

hh <- subset(hh,hh[,1]>=t1 & hh[,1]<=t2 & hh[,5]!=-9.99 & hh[,5]>=3) #Sub-set data period

#Sub-set data for labelling
hh1 <- if(nrow(subset(hh,hh[,5]>=5.2))>0) {subset(hh,hh[,5]>=5.4)} else
{subset(hh,hh[,5]>=floor(range(hh[,5])[2]))}

#Remove all objects aside from specified data.frames
rm(list=ls()[!(ls()%in%c("t1","t2","t3","t4","dd","hh","hh1"))])

#Data plotting
lab1 <- pretty(range(na.omit(dd[,c(5,7)])))
lab2 <- seq(min(lab1),max(lab1),diff(lab1)[1]/2)

lab3 <- pretty(range(hh[,5]))
lab4 <- seq(min(lab3),max(lab3),diff(lab3)[1]/2)

#Set-up device

```

```

ii <- "../[...]/imo_logo.gif" #VÍ logo
jj <- "../[...]/hi_logo.png" #HÍ logo
kk <- "../[...]/avd_logo.jpg" #RLS logo
ll <- "../[...]/lhg_logo.jpg" #LHG logo
mm <- Sys.time()
nn <- "barc_gps_all_is.pdf"
oo <- noquote(paste(gsub("\\.pdf", "", nn), ".png", sep=""))

setwd("../[...]/") #[...]vedur.is

pdf(file=nn, title="matthew@vedur.is",
    paper="special", width=12, height=8, bg="white", family="Helvetica")

par(oma=c(6, 4, 5, 6)) #Outer margins
par(las=1)

#Uppermost graph
par(mar=c(0, 0, 0, 0)) #Plot margins (no. of lines). Base; left; top; right
par(fig=c(0, 1, 0.25, 1)) #25-100% of height of fig. region: c(x1, x2, y1, y2)

plot.default(dd[, 1], dd[, 5], type="n", col="red", lwd=1.2, lty=1, las=1,
    xaxs="i", yaxs="r", xlim=range(t1, t2),
    ylim=range(lab1), ann=F, axes=F, xaxt="n", panel.first=grid(NA, NULL, col="black", lwd=0.8, lty=3))
lines(dd[, 1], dd[, 5], type="l", col="lightgrey", lwd=1.2, lty=1)
lines(dd[, 1], dd[, 7], type="l", col="blue", lwd=1.5, lty=1)
axis.POSIXct(1, range(t1, t2), at=seq(t1, t2, by="2 days"), "", labels=F, tcl=-0.3)
axis.POSIXct(1, range(t3, t4), at=seq(t3, t4, by="8 days"), "", labels=F, tcl=-0.5)
axis(2, at=lab1, labels=T, tcl=-0.5)
axis(2, at=lab2, labels=F, tcl=-0.3)
axis(4, at=lab1, labels=T, tcl=-0.5)
axis(4, at=lab2, labels=F, tcl=-0.3)
abline(v=c(hh1[, 1]), col="firebrick1", lwd=1.3, lty=2)
abline(v=mm, col="darkgrey", lwd=1.2, lty=5)
text(as.numeric(unclass(mm))+253800, lab1[ceiling(length(lab1)/2)],
    labels=paste(format(mm, "%H:%M"), "UTC"), col="black", cex=1.2, srt=90)
par(las=0)
mtext("Löðrétt færsla (m)", side=4, line=4, adj=0.5, col="black", cex=1.2)
par(las=1)
legend("topleft", legend=c("Mismunagildi", "180-minútna miðgildi"),
    xjust=0.5, yjust=0.5, lwd=2, lty=1, col=c("lightgrey", "blue"), bg="white", ncol=2, cex=1)
box()

#Lowermost graph
par(mar=c(0, 0, 0, 0)) #Plot margins (no. of lines). Base; left; top; right
par(fig=c(0, 1, 0, 0.22), new=T) #0-22% of height of fig. region: c(x1, x2, y1, y2)

plot.default(hh[, 1], hh[, 5], type="n", col="black", lwd=0.8, lty=1, las=1, yaxs="r", xlim=c(t1, t2),
    ylim=range(lab3), ann=F, axes=F, xaxt="n", panel.first=grid(NA, NULL, col="black", lwd=0.8, lty=3))
points(hh[, 1], hh[, 5], pch=22, lwd=0.8, lty=1, col="black", bg="white", cex=1)
text(as.POSIXct(hh1[, 1]), hh1[, 5], labels=round(hh1[, 5], 1), pos=c(2, 3, 4),
    offset=0.2, col="firebrick1", cex=1) #Label events
axis.POSIXct(1, range(t1, t2), at=seq(t1, t2, by="2 days"), "", labels=F, tcl=-0.3)
axis.POSIXct(1, range(t1, t2), at=seq(t3, t4, by="8 days"), "", labels=F, tcl=-0.5)
axis.POSIXct(1, range(t3, t4), at=seq(t3, t4, by="8 days"),
    "%d/", labels=T, tcl=-0.5, line=-0.35, lty=0)
axis.POSIXct(1, range(t3, t4), at=seq(t3, t4, by="8 days"),
    "%m/%y", labels=T, tcl=-0.5, line=0.5, lty=0)
axis(2, at=lab3, labels=T, tcl=-0.5)
axis(2, at=lab4, labels=F, tcl=-0.3)
axis(4, at=lab3, labels=T, tcl=-0.5)
axis(4, at=lab4, labels=F, tcl=-0.3)
abline(v=mm, col="darkgrey", lwd=1.2, lty=5)
par(las=0)
mtext(expression(paste("Stærð(", italic(M)[lw]>=3, ")")),
    side=4, line=4, adj=0.5, col="black", cex=1.2)
par(las=1)
box()

title(main="Hæðarbreyting í öskju Bárðarbungu", outer=T, line=3, col="black", cex.main=1.5)
mtext("Löðrétt færsla fengin með rauntímaúrvinnslu úr GPS tæki staðsettu í miðri
öskjunni", outer=T, line=1.6, col="black", cex.main=1)
mtext(paste("Uppfært:", format(Sys.time(), "%d.%m.%Y %H:%M
UTC")), outer=T, line=0.5, col="black", cex=1)
mtext(paste("Hæð:", round(head(dd[, 2], n=1), 1), "m"), outer=T, adj=-
0.05, line=0.3, col="black", cex=1)

```

```

mtext("Mælingar eru samvinnuverkefni HÍ, VÍ, Almannavarna og Landhelgisgæslunnar",outer=T,
adj=-0.05,line=-33,col="black",cex=1)
mtext("Hæðartölur miðast við hæð yfir sjó",outer=T,adj=1.09,line=-33,col="black",cex=0.8)
mtext(bquote(italic(.paste("Gögnin eru sýnd hér í almannavarnaskyni og til upplýsingar fyrir
almenning og vísindamenn. Ekki er heimilt að nota gögnin til vísindalegrar útgáfu án
leyfis."))))),outer=T,adj=0,line=-34.2,col="black",cex=0.8)
mtext(bquote(italic(.paste("This graph is for civil protection use and to inform the public.
It is not to be used in scientific publications without permission."))))),outer=T,adj=0,line=-
35,col="black",cex=0.8)

dev.off()

rm(list=ls()[!(ls() %in% c("ii", "jj", "kk", "ll", "nn", "oo"))])
gc() #Garbage collection - return some memory to OS

system(paste("nice /[.]/pdftoppm -r 100 -png ",nn," > ",oo,sep=""),wait=T) #PDF to PNG format
system(paste("/[.]/composite -compose atop -gravity NorthEast -geometry
+240+10",ii,oo,oo,sep=" "),wait=T) #Overlay VÍ logo
system(paste("/[.]/composite -compose atop -gravity NorthEast -geometry
+160+10",jj,oo,oo,sep=" "),wait=T) #Overlay HÍ logo
system(paste("/[.]/composite -compose atop -gravity NorthEast -geometry
+80+10",kk,oo,oo,sep=" "),wait=T) #Overlay RLS logo
system(paste("/[.]/composite -compose atop -gravity NorthEast -geometry
+10+10",ll,oo,oo,sep=" "),wait=T) #Overlay LHG logo

#Copy PNG image to the web server
system(paste("/[.]/scp ",oo," ", "[.]/",sep=""),wait=T)
system(paste("/[.]/scp",oo," [.]/"),wait=T)

rm(list=ls()) #Remove objects from working environment
quit("no")
#EOF

```