University of Cagliari

Doctoral Thesis

International PhD in Environmental Science and Engineering
XXV Cycle

Implementation and management of an Environmental Data System for climatological research

Author:
Filippo Locci

Supervisor:
Dr. Maria Teresa Melis

PhD Coordinator:
Prof. Roberto Orrú

TeleGIS Laboratory
Department of Chemical and Geological Sciences

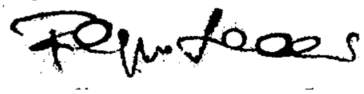
Academic year 2012-2013

Declaration of Authorship

I, Filippo Locci, declare that this thesis titled, 'Implementation and management of an Environmental Data System for climatological research' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:



Date: 28-04-2014

"Ringrazio mia moglie Silvia, che in questi anni mi ha sempre incoraggiato e sostenuto nel portare avanti questo lavoro"

Filippo

UNIVERSITY OF CAGLIARI

Abstract

Faculty Name
Department of Chemical and Geological Sciences

Doctor of Philosophy

Implementation and management of an Environmental Data System for climatological research

by Filippo Locci

The implementation of an Environmental Data System (EDS) for the Share Geonetwork web-platform dedicated to the management and sharing of climatological data acquired from sensors, stations and from physical or chemical analysis from single researcher or scientific institution is the main issue of this PhD research activities. Purpose of this research is:

1. the implementation of the EDS for a webGIS service platform dedicated to the management and sharing of climatological data acquired by high elevation stations (SHARE GeoNetwork project, promoted by the Ev-K2 CNR Committee) and data derived from chemical and physics characterization about non-polar ice cores and marine sediment core ("Project of Strategic Interest - PNR 2011-2013 NextData.").
2. the development of an automatic system useful to validate MODIS product over Hymalaya using Ev-K2 CNR Committee high-altitude data (presentation of a case study).

1. The web-platform will provide basically three types of services:
   (a) structured metadata archive, data and results from high-altitude and marine environments researches and projects;
   (b) access to meteo-climatic and atmospheric composition data, past climate information from ice and sediment cores, biodiversity and ecosystem data, measurements of the hydrological cycle, marine re-analyses and climate projections at global and regional scale;

(c) dedicated webGIS for georeferenced data collected during the research.

2. In particular the elaboration process is performed by the use of Python script on the MODIS/Terra Snow Cover Daily L3 Global 500m Grid (MOD10A1) data and metadata (snow cover, snow albedo, fractional snow cover, and Quality Assessment (QA) data in compressed HDF-EOS format) and radiative data (Short (SWR) and Long (LWR) Wave Radiation) from Nepal Climate Observatory-Pyramid (NCO-P) .

Methods

1. High elevation environmental and territorial data and metadata are cataloged in a single integrated platform to get access to the information heritage of the SHARE project, using open source tools: Geonetwork for the metadata catalog and webGIS resources, and the open source Weather and Water Database (WDB), developed by the Norwegian Meteorological Institute, for the database information system implementation. A specific database, named WDBPALEO, formed by IDB (Ice Core Database) and SDB (Sea-Core Database), has been developed for the collection of paleoclimatological data from ice cores extracted from glaciers in non-polar ice cores and marine sediment cores. The storage structure of the data is based on the architecture of WDB. In the first phase of the project, the database WDB has been modified to input data from high altitudes meteorological stations in Nepal, Pakistan, Africa and Italy. Moreover, the need to have a tested database able to contain data of different nature has led to the modification of the program source to store data of different nature and with unlimited temporal extension such as ice and sediment cores.

2. For data elaboration and comparison a temporal scale of two months has been used. NCO-P records SWR and LWR (both upwelling and downwelling, useful to calculate daily short wave albedo) since 2003. Corresponding time interval of MOD10A1 data have been downloaded from http://reverb.echo.nasa.gov/reverb. From Pyramid data Short-Wave albedo has been calculated for each day of the pre-monson period (April and May) and compared with the mean value of 4 neighborhood snow albedo pixels where the Pyramid station is located.

Results and Conclusions

1. The tool that is being studied and developed proposes an environment for the research, dedicated mainly to high-altitude studies, allowing researchers to access the wealth of resources that are acquired by the climatological stations. With

regards to remote sensing data processing, the system gives the opportunity to free access to the data relevant either for atmospheric corrections or ground data calibrations. Moreover, into SHARE GeoNetwork, a specific topic is dedicated to the description of the new technologies developed in the SHARE Project. The core of the system is the catalog of metadata based on GeoNetwork open source application, proposed on the principles of Free and Open Source Software (FOSS) and International and Open Standards for services and protocols (from ISO/TC211 and OGC). A new hierarchical structure has been implemented into SHARE for the description of metatada of the stations, sensors and measures. Also , following the scope of the NEXTDATA project a methodology to recover, store, access and disseminate data derived from chemical and physics characterization about non-polar ice cores and marine sediment core has been developed. They give precious information about the evolution of anthropogenic pollution, climate variability and about the composition of middle troposphere. With an accurate bibliography research we managed to collect a great amount of ice core and sediment core data and metadata that were essential to study a suitable methodology to reach this goal. Data collected in the spatial geographic database were integrated in an information system through specific web services. The first release of SHARE Geonetwork is working from 2011 and it is accessible for public at http://geonetwork.evk2cnr.org. Second release, recently published is accessible at: http://geonetwork.evk2cnr.org/index.php, with data navigation, visualization and dowload.

2. The Python code, with Modis reprojection tool, collected and modified should be useful to manage and elaborate a large amount of data, giving a clear methodology for the validation of MODIS data and products, in particular using Open Source tools. A moderate correlation exists (0.50) between the calculated albedo and the MOD10A1 albedo product. In order to improve the results of this study the data were subdivided into three groups on the base of the acquisition date. The correlation between these new groups of data appear increased, guessing the solar zenith angle influences in the MODIS snow albedo retrieval. It will be necessary further statistical evidences to confirm this correlation.

Acknowledgements

This work has been developed in the framework of three projects: SHARE, Nextdata and I-Amica all financed by MIUR. My thanks to Ev-K2 CNR Committee that gave to me the opportunity of this work and for the use of its data for my research activities. My special thanks for my advisor Maria Teresa Melis and my collegue Francesco Dessì for the trust given to me and for their support. Thanks to the TeleGis Laboratory and the Department of Chemical and Geological Sciences in the person of Professor Giorgio Ghiglieri. Finally I intend to thank the PhD coordinator Professor Roberto Orrù and Professor Giacomo Cao for their support.

Contents

Declaration of Authorship **i**

Abstract **iii**

Acknowledgements **vi**

Contents **vii**

List of Figures **ix**

List of Tables **xi**

1 Introduction **1**
1.1 The Environmental Data System 1
1.2 SGN Environmental Data for climatological studies 1
1.2.1 High Altitude data 1
1.2.2 The SHARE Project 2
1.2.3 Non-polar glaciers and ice core 3
1.2.4 Marine sediment cores 4
1.2.5 The NextData Project 4
1.3 Available data repositories: ice core and sediment core data 4
1.3.1 Marine Sediment core data repository 5
1.3.2 Non Polar Ice core data repository 6
1.4 Thesis structure 7

2 SHARE Geonetwork and Weather and Water Database (WDB) **8**
2.1 GeoNetwork Background 8
2.1.1 GeoNetwork in the framework of the SHARE Project 9
2.1.2 Interactive maps 10
2.2 The Network and sensors 11
2.2.1 Network description 11
2.2.2 Data Validation 13
2.3 Database Implementation 14
2.3.1 Material and Method 15
2.3.2 WDB adaptation 16

2.3.3 Data loading 20
2.4 Web queries from SHARE Geonetwork website 21

3 The paleoclimatology database: WDBPALEO 27
3.1 Introduction 27
3.2 Main features of the WDBPALEO system 29
3.3 Time field adaptation 32
3.4 WDBPALEO data loading 32
3.4.1 WDBPALEO-IDB characterization 32
3.4.1.1 Reference system 32
3.4.1.2 Structures adaptations 33
3.4.1.3 WDBPALEO-IDB data loading 33
3.4.2 WDBPALEO-SDB characterization 40
3.4.2.1 WDBPALEO-SBD main adaptations 41

4 A comparison between MODIS Albedo Snow Product and in situ SHARE observations over the Himalaya Region: a case study 44
4.1 Introduction 44
4.2 Relevant SHARE AWS data for remote sensing application 44
4.3 Presentation of the case study 45
4.3.1 The MOD10A1 products overview 45
4.3.2 Total shortwave albedo 46
4.3.3 Study area 48
4.4 Material and methodology 48
4.5 Results 60

5 Discussion and conclusions 64
5.1 Introduction 64
5.2 The EDS: a critical overview and future perspectives 64
5.2.1 First conclusions 67
5.3 EDS's data use: discussion about results and methodology 68

A SHARE Network Measuring Sensors 70

B WDBPALO Scripts 72

Bibliography 80

List of Figures

2.1 Metadata . . . 10
2.2 Nepal Monitoring Network . . . 13
2.3 Data Flow . . . 17
2.4 SCRIPT1 . . . 19
2.5 SCRIPT2 . . . 19
2.6 SCRIPT3 . . . 20
2.7 WRITE . . . 20
2.8 AWS2WDBF . . . 21
2.9 DIAL2 . . . 22
2.10 SGNHOME . . . 23
2.11 Login . . . 24
2.12 GDJ-Fig1 . . . 24
2.13 PHP . . . 24
2.14 GDJ-Fig4 . . . 25
2.15 GDJ-Fig5 . . . 25
2.16 RAWDATA . . . 26

3.1 POINT DATA COMPARISON . . . 27
3.2 WDBPALEO SYSTEM . . . 28
3.3 Sample ice core analysis . . . 28
3.4 WDB DATA MODEL . . . 29
3.5 WDBPALEO schema sample . . . 30
3.6 WDB global schema . . . 31
3.7 wdb_int.icecorename . . . 33
3.8 wdb_int.floatvalue-idb . . . 34
3.9 Taldice cutting plan . . . 38
3.10 Percentage and type of data stored . . . 39
3.11 WDBPALEO-IDB ER schema . . . 40
3.12 SDB initialization . . . 42
3.13 Benthic Foraminifera parameters adding script . . . 43
3.14 Dating measurement table . . . 43
3.15 wci.addcoredatingdatingmeasurement example of use . . . 43

4.1 NCOP . . . 49
4.2 REVERB dowload . . . 50
4.3 Accumulated SW albedo . . . 52
4.4 GEOTIFF images . . . 54
4.5 Extracted Snow Albedo data . . . 59

4.6 Snow albedo map 60
4.7 Snow albedo vs MOD10A1 albedo scatter plot 63

5.1 SGN resources access 66
5.2 SHARE Geonetwork user hierachy 67
5.3 Statistical albedo correlation analysis 69

A.1 AWS Meteorological Paramameters 70
A.2 AWS Atmospheric Paramameters 71

B.1 WDBPALEO Loading Data Script 1 73
B.2 WDBPALEO connecting Script 74
B.3 WDBPALEO Loading Data Script 2 75
B.4 WDBPALEO-IDB wci.writepaleo 76
B.5 WDBPALEO-SDB wdb_int.coredatingmeasurement 77
B.6 The WDBPALEO-SDB function wci.writepaleo 78
B.7 The WDBPALEO-SDB view wci_int.floatvalue_v 79

List of Tables

2.1 SHARE Network - AMS (Atmospheric monitoring station), AWS (Automatic weather stations) 12
2.2 Data dimensions 18

3.1 Dataprovider stored in to IDB database. 35
3.2 Dataprovider stored in to IDB database 37
3.3 List of some parameters and measurements unit in the IDB database. . . 38

4.1 The MOD10A1 SDSs, (A. George et al, 2006) 47
4.2 Local Attributes for Snow Cover and Snow Albedo 55
4.3 Statistics 62

For my wife Silvia, and my sons Samuele and Davide

Chapter 1

Introduction

1.1 The Environmental Data System

The scope of this dissertation is to show the design, the development and the implementation of the Environmental Data System (EDS) for the SHARE Geonetwork Web-Platform (SGN), in the framework of SHARE and NextData projects. The EDS is designed to store data from two sources: high altitude automatic stations, non-polar ice cores and marine sediment cores. SGN has the daunting objective to become a Data Assembly Center dedicated to the collation, archival and sharing of environmental data and products.

1.2 SGN Environmental Data for climatological studies

The SGN system has been designed to give access to: meteoclimatic and atmospheric composition data, past climate information from ice and sediment cores, biodiversity and ecosystem data, measurements of the hydrological cycle, marine reanalyses and climate projections at global and regional scale. All of this data and elaborations are the result of measurement from networks of observations in remote mountain and marine areas.

1.2.1 High Altitude data

Mountain areas are an important factor in the climatic system, and they are one of the trigger mechanisms of cyclogenesis in mid latitudes, through their perturbations of large-scale atmospheric flow patterns. They also have an influence on the formation of clouds and precipitation in their vicinity, which are in turn indirect mechanisms of heat and moisture transfer in the vertical. Consequently, the influence of orography on climate needs to be taken into account in a physically- meaningful manner. (**?**). Hence, high mountains provide information that tell us about climatic changes such as melting glaciers and rainfall pattern changes (UNEP, 2009). Monitoring, collecting information and data elaboration from high mountain sites is fundamental to the preservation of mountain ecosystems. The development of a high mountain meteorological and atmospheric network is essential to monitor the climate change impact on its ecosystem and the resulting feedback on the cryosphere (Grabherr et al., 2000). Nevertheless, many networks which study climate changes do not include strategic high altitude locations around the world due to the difficulties in the management of such sites and the only recent understanding of their scientific importance. Pursuing the purpose of safeguarding the environment, since 1987 the Ev-K2 CNR Committee, an autonomous non-profit association, promotes scientific and technological research in mountain areas accepting the challenge to fill the scarcity of data collected in mountain regions, in spite of the extreme topography and harsh climate conditions that are at the main reason for this data gap. Nowadays, the most important monitoring networks, operating around the world that collect climate, atmospheric and terrestrial data, are developed and maintained in the context of international project such as: Atmospheric Brown Clouds (ABC), Aerosol RObotic NETwork (AERONET), Coordinated Enhanced Observing Period (CEOP), Global Atmosphere Watch (GAW), International Long-term Ecological Research Network (ILTER), Global Land Ice Measurement from Space (GLIMS), Global Seismographic Network (GSN), International GNSS (Global Navigation Satellite Systems) Service (IGS). Some of the Ev-K2 CNR network sites (Pyramid Laboratory - Observatory in the Khumbu Valley, and Italian Climate Observatory "Ottavio Vittori" at Mount Cimone) are included in these international project: ABC, AERONET, CEOP and GAW. Since 2005, the Ev-K2 CNR Committee has promoted an integrated environmental project named SHARE (Station at High Altitude for Research on the Environment) focused on the mountain regions as primary indicators of climate change. Originally launched as a system of measurements in environmental and earth sciences in the Himalaya-Karakorum region, SHARE has later expanded its network to Europe (Apennines and Alps), Africa (Rwenzori) and more recently to South America (Andes).

1.2.2 The SHARE Project

The specific aim of SHARE is improving scientific knowledge on climate variability in mountain regions, ensuring the availability of long term and high quality data concerning atmospheric composition, meteorology, glaciology, hydrology, water resources, biodiversity and human health. SHARE activities also plan to include the design of mitigation and adaptation strategies to oppose the effects of climate change. The SHARE project is divided in four work packages and includes the following sectors: scientific research and climate, technological research and climate, information system, and capacity building. In particular, within the SHARE project, main goal of information system work package is to implement an High Mountain's Data Information System able to respect the international standards and giving to the scientific community a resource in term of climate change understanding.

1.2.3 Non-polar glaciers and ice core

In terms of Climatic Change response non-polar glaciers are the most sensitive element of the cryosphere. This response faster and more intense than polar glaciers or polar ice caps (Lemke et al., 2007). Changes in mountains and in particular in non-polar glaciers can produce strong effects on water resources that can impact agriculture, electricity production, urban ecosystem and on the communities living in these areas. Each variation that can occur in snow precipitation and ice cover might generate a significant impact on the fluvial basin, not only on the variation of the water flow but also on the potential increase of flooding, erosions and the related increase of the risk (Beniston et al., 1997). Therefore, ice cores could be considered the best indicators of climate variability (Legrand et al., 2002). During snow precipitations snowflakes store information about the atmospheric state at the moment of its formation. In this way, accumulation of snow during one or several years produces a detailed archive of the chemical composition and physical condition of the atmosphere at the precise moment of freezing (Paterson and Waddington, 1984). Tropical, subtropical and mid-latitude glaciers offer unique information about climate thanks to the fact that snow precipitation is their unique growth factor. The glaciers store high resolution information about the temporary climate, especially in areas that do not present appropriate past data and information about natural climate alteration processes or the anthropic pollution evolution. Few areas, on mid-latitude, sub-tropical and tropical mountain chains permit snow conservation and glacier formation over the years. Thus, only above 4000 m a.s.l. there are the climate conditions that allow the conservation of snow. The mid-latitude glaciers can offer unique, useful and complementary information that ice caps can't provide (Delmas, 1992). Snow accumulation represent a unique natural archive for the anthropogenic impact history and meteorological conditions. From this point of view detailed information about mountain regions allow us to determine the evolution of the local climatic system in the recent past and of the stochastic evolution that can occur in the next few decades (Thompson et al., 2000). Time series of environmental changes are extracted by analyzing ice cores and boreholes. Useful information deriving from many of these patterns is preserved not only in the ice sheets, but also in ice caps in Arctic Canada, Alaska, Patagonia and in high-altitude glaciers in Tibet, Andes, and other mountainous regions (Paterson and Waddington, 1984).

1.2.4 Marine sediment cores

Like ice cores a marine sedimentary cores give a lot of information about the processes that drive the climate-subsystem: lithosphere (sedimentary regime), biosphere (productivity, plankton and benthos response), oceans (circulation and water oxygenation, ocean ventilation) and atmosphere (eolian input fluctuations to marine basins) (Lirer et al., 2013). Accordingly, numerous climatic evolution studies in marine sedimentary records have been carried out in the last decade starting from the availability of long time series describing paleoclimatic variability with different time resolution (Lemke et al., 2007). Periodical (in several time scales) changes in atmospheric and oceanic temperatures (both in global scale and in regional scale) during time (Ruddiman, 2001) are pointed out by climate proxies. In this way marine sediment core allow us to have undisturbed records (continues and with high sedimentation rate) ((Dansgaard et al., 1993), (Bond et al., 1997), (Shackleton et al., 2000), (Mayewski et al., 2004)) and contain a summary of all processes involved in sediment origin (Hayes, 1999). Useful information to understand current patterns of climate variability could be derived using the wide range of geological time scales and the different spatial sensitivity. This means that highly sensitivity regions ensure more paleoclimatic information, such as the Mediterranean Sea. The Mediterranean could be considered an area where the response to climate variation is amplified due to its morphology. (Lirer et al., 2013).

1.2.5 The NextData Project

The work carried out in this research is a part of NEXTADA, an Italian project which aims to structure an implementation of measurement networks in remote mountain and marine areas. The NEXDATA project will develop efficient web portals to access meteo-climatic and atmospheric composition data, past climate information from ice and sediment cores, biodiversity and ecosystem data, measurements of the hydrological cycle, marine re-analyses and climate projections at global and regional scale.

1.3 Available data repositories: ice core and sediment core data

Nowadays there exist a long list of data repository to store data, in particular environmental data, even so in this section will be given in consideration ice core and sediment core data repository. For completeness, however, it is good to mention data repository for data acquired from Automatic Weather Stations (AWS), such as observations. This repository is usually a prerogative of Meteorological or Atmospheric Centers all over the world (Governmental or Non Governmental Centers). Among these, noteworthy is The Australian Bureau of Meteorology with the section "Climate Data Online" (http://www.bom.gov.au/climate/data/?ref=ftr) dedicated to query the Bureau climate dataset. The user can query (in text or using map) free all the Automatic Weather Station of the Bureau for the temperature, rainfall and solar exposure time series, obtaining graphs and statistics (other data request are available with fee).

1.3.1 Marine Sediment core data repository

At present, the main repository for Marine Sediment Core Data is The Index to Marine and Lacustrine Geological Samples (IMLGS), belonging to the NOAA's National Geophysical Data Center (NGDC), in the section Marine Geology and Geophysics. The IMLGS is a tool to help scientists to locate and obtain geologic material from sea floor and lakebed cores, grabs, and dredges archived by about 30 participating institutions around the world. The Index to Marine Geological Samples began in 1977 as an 80-column coding form using 1-digit codes to represent common vocabularies designed and agreed upon by the participating institutions. This explains some non-normal aspects of the database design, and also the extensive uppercase abbreviations in the comments field of many records. Over the years, the database evolved to a 95-column form, then to a relational database. The 1-digit codes remain only for reference in associated tables, paired with the plain-English terms used as foreign key constraints to enforce common vocabularies. The IMLGS now uses the vocabularies, rather than the 1-digit codes. They have experimented with serving the vocabularies as an RDF graph, but have limited resources and the effort is on hold. Originally, the IMLGS tables used a compound primary key consisting of ship/cruise/sample/device - used in the 1980s to enforce unique sample identifiers. An "imlgs identifier" (generated on insertion) as a primary key is used. The IMLGS strength is maintaining the collaboration with the IMLGS partner repositories, and that it links to many related resources of use to researchers looking for geosample material. Its contents are definitely better than its structure and implementation. The International GeoSample Identifiers (IGSNs) wherever known are incorporated, and cross-link to/include Rolling Deck to Repository (R2R) cruise identifiers. The IMLGS database has been developing in Oracle, with an instance at R2R in Postgres, sending database updates to R2R regularly. A variety of web services and an ArcGIS web interface are available from NGDC, based on a geospatially-enabled Oracle table. R2R also serves the IMLGS as a virtual RDF (Resource Description Framework) graph from their Postgres copy using a system for the relational data access as virtual (D2RQ Platform http://d2rq.org/). Data search can be performed in two ways: by interactive map and by text. The text search interface (at http://www.ngdc.noaa.gov/geosamples/) is a simple JSP interface written using JSTL. NGDC is phasing out this sort of interface in favor of enterprise solutions using Grails. The ArcGIS web interface uses ESRI technology, and the small maps on the individual institutional web pages are implemented using OpenLayers. IMLGS is a type of irreplaceable repository to have access to massive paleodata resources. Among IMLGS partners, the Lamont-Doherty Earth Observatory at Columbia University as part of the Integrated Earth Data Applications (IEDA) data facility has been developed and maintain SedDB. SedDB is a data management and information system for marine sediment geochemistry. It provides access to data of fundamental interest to a wide range of research topics, from paleoclimate reconstructions to fluxes between the Earth's surface and mantle (http://www.earthchem.org/seddb). A text interface is the only way to query SedDB. In terms of access to the resources whole repositories give data in bulk form. This is a file containing geosample data related to the selected object (core, reference, ship etc.).

1.3.2 Non Polar Ice core data repository

The NOAA National Climatic Data Center (NCDC) (NOAA, 2013) and the National Ice Core Laboratory (NICL, 2009) are the main repositories of non polar ice core data. Data ice core analysis, on a global scale, are stored and available for download in .txt or .xls formats, whose content is two-fold: metadata about the principal research's investigator and the journal where data is published; data about chemical and physical analysis of ice core samples. In the NOAA-NCDC website the Ice Core Gateway includes "The World's Ice Cores", which is a list compiled by the International Ice Cores Data Cooperative of some ice cores. The NOAA-NCDC gateway classifies Ice Cores in 5 subgroups: Antarctica, Greenland, Other Polar Ice Core, Tropical and Temperate Core, Sea Ice Core. The strength of the NOAA-NCDC is that it provides in a single way the NOAA's Paleoclimatology Program ice core data archives. However, the data is available in a bulk format only. The NCDC's database is not designed to search a single parameter and its associated value for an ice core data analysis. This is a limitation for the user who needs to check a specific parameter. The U.S. National Ice Core Laboratory (NICL) facility is designed for storing, curating, and studying ice cores acquired from polar and non-polar regions. It provides scientists with the ability to conduct examinations and measurements on ice cores, and it preserves the integrity of these ice cores in a long-term repository for current and future investigations. Responsibility for sample allocation falls under the NICL Science Management Office. Distribution policies are available at `http://nicl-smo.unh.edu/access.html`. The NICL database is only a table with different rows and columns filled with data, which can be downloaded from the site.

1.4 Thesis structure

The present dissertation is composed of 5 chapters, through which the main objective of this research is explained and studied in depth.

- Chapter 1: Introduction.
  In the first Chapter the objective of the research and the context of the research are defined. In particular the environmental data type involved in the EDS design. At the same time the main environmental data repositories for marine sediment cores and non-polar ice cores are taken into account.

- Chapter 2: SHARE Geonetwork and Weather and Water Database (WDB).
  This Chapter gives an in depth survey of the SHARE Geonetwork System: metadata catalog and its implementation and the database system adaptation and implementation starting from the WDB system (Locci et al., 2014).

- Chapter 3: WDBPALEO.
  Chapter 3 illustrates the new paleoclimatological database named WDBPALEO which is designed modifying the WDB's source code and its structure.

- Chapter 4: A method for Remote Sensing data validation using SHARE data.
  This Chapter is the result of a work presented at the First International Conference on Remote Sensing and Geoinformation of Environment (Melis et al., 2013). This work described a methodology for the comparison of MODIS Snow Products (MOD10A1) with radiative data observations from the Pyramid Station.

- Chapter 5: Discussion and conclusions

Chapter 2

SHARE Geonetwork and Weather and Water Database (WDB)

The Share Geonetwork platform for web services is based on the architecture of GeoNetwork Open Source for the development of the data and metadata catalog dedicated to the high altitude research related to the SHARE project.

2.1 GeoNetwork Background

A prototype of the GeoNetwork catalogue was developed by the Food and Agriculture Organization of the United Nations (FAO) in 2001 to systematically collect and publish the geographic datasets produced within the organization. Moreover, the World Food Programme (WFP) joined the project and with its contribution the first version of the software was released in 2003. GeoNetwork has been developed following the principles of Free and Open Source Software (FOSS) and based on International and Open Standards for services and protocols, like the ISO-TC211 and the Open Geospatial Consortium (OGC) specifications (OCG, 2012). Currently, there is a big community that uses and develops the GeoNetwork software. GeoNetwork Open Source consists of:

- advanced search discovery tools,
- metadata catalogue, descriptive records for dataset,
- geographic data and map viewer.

Improving sharing and access to data and information by the use of this open source system it is possible, also adopting two international standards: DUBLIN CORE for general documents and ISO19139 for geographic data. The use of these standards ensure that metadata can be interpreted by software and users and make data and information resources easier to find through the world wide web. In terms of Geonetwork competitors, there are a few open source projects (e.g. CKAN, geOrchestra, Mdweb, pycsw) that provide metadata through the OGC CSW (Catalog Service for the Web) interface. A couple of proprietary metadata catalogs project focus on very specific user communities like earth observation. The main advantages of GeoNetwork over those catalog applications are: its huge user community; reliability; performance, compliance and metadata management functions that have all been tested and used in operational systems for years; in addition offers more standard supports. Also, Geonetwork team programmers developed an extension to Esri ArcGIS Desktop named GeoCat Bridge to transfer your geospatial data from the desktop to a Spatial Data Infrastructure. It will create the required map services on GeoServer and also in MapServer. Data is transfered to the server platform, storing it on disk or loading it into a PostGIS database. Styling is taken from the ArcGIS project, generating the SLD documents. Metadata is also maintained at the source and transformed into valid ISO 19115 metadata that is published in a GeoNetwork catalog. Esri released their Geoportal toolkit as an open source product in 2010.

2.1.1 GeoNetwork in the framework of the SHARE Project

The first phase of the SHARE project was dedicated to the completion of the cataloging system of climate observatories and weather stations in high mountain regions included in the program, following guidelines on the use of metadata for WMO Information System (Tandy, 2010). In the first phase a metadata catalogue containing all the data collected has been developed and made available because:

- metadata is primary to developing any digital collection of files,
- metadata are important for resource discovery and use,
- several metadata standards or schemes are used to aid authors to assign metadata to their files.

The new SHARE metadata catalog allows stakeholders to find if a data set exists, to identify a contact person for data of interest and to download actual data from the GeoNetwork node if ownership rights permit. The SHARE-Geonetwork provides the connection with the high mountain stations of the SHARE network, giving a new

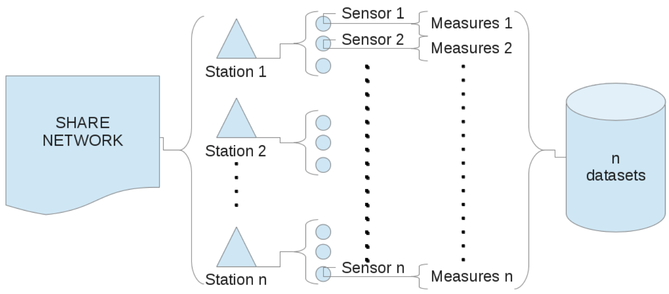


FIGURE 2.1: Metadata hierarchy

category of data resources. Regarding the metadata of the stations, there is an important improvement of the catalog due to hierarchical organizations that has been proposed and developed. The stations metadata hierarchy can be synthesized by the model in Fig. 2.1. Each element of the chain depends on the element from which it derives and generates dependency through logical links. In this model the network of SHARE stations is the upper hierarchical unit, the first series has its own metadata. Each station has own metadata: abstract, purpose, points of contact, geographic location. Each station is equipped with its own sensors generating measures, also described by the metadata in the form of series and dataset-level that describes the data. Metadata are updated whenever new data are loaded or a change on data, on sensors, on instruments and in information related on data occurs. Also, new metadata scheme are developed whenever new type of environmental data need to be added to the web-service platform.

2.1.2 Interactive maps

Geonetwork open source provides Internet access to interactive maps, satellite imagery and related spatial databases with the possibility of layers management and queries through an integrated WebGIS based on GeoServer software (Pumphrey, 2009). The integration of Google Maps cartographic data in the GeoServer has been developed making available advanced and complete satellite images, street maps, physical maps and hybrid maps. The system was deeply modified to bring the configurations to the publication of the Google's layers: the main issue is related to the change of the cartographic parameters to make them compatible with the publication in Google. The interface of Geonetwork SHARE was then changed in the configuration files and scripts with their dependencies that concern the segment mapping.

2.2 The Network and sensors

The SHARE environmental monitoring system, originally launched as an integrated system of measurements in environmental and Earth sciences in the Himalaya - Karakorum region, SHARE increasingly responds to an international call for information on the effects of climate change. According to a 2007 IPCC report (Lemke et al., 2007), worldwide climate change threatens ecosystem balance and, as a consequence, life on Earth. SHARE Network which specializes in collecting important but otherwise unobtainable data at high altitudes, contributes information to several of the above networks. Particularly requested is information on meteo-climate parameters, atmospheric chemical measurements, limnological and paleolimnological analysis on high altitude lakes, glaciological data and precise measurement of Earth surface coordinates. To better respond to the global environmental problems, SHARE has expanded its network beyond Asia (Nepal and Pakistan) to include stations in Africa (Uganda) and Europe (Italy and other EC Countries), with expansion to South America in the planning stages. (http://www.evk2cnr.org/cms/en/research/integrated_programs/share)

2.2.1 Network description

The SHARE monitoring network involves fifteen stations (automatic weather stations and laboratories) spread across three continents and four countries (Italy, Nepal, Pakistan and Uganda, see Table 2.1).

The majority of installations are in emerging countries where they support local governments in decisions regarding environmental subjects and sustainable development policies. In Italy, the observation network includes four points: the Italian Climate Observatory "O. Vittori" (ICO-OV) on Mount Cimone, the Forni Glacier AWS, the Dosdè Glacier AWS and the Gigante Glacier AWS. The installation site at Mount Cimone, the highest peak of the Italian Northern Apennines, is managed by the Institute of Atmospheric Sciences and Climate of the Italian National Research Council (ISAC-CNR), and hosted within the infrastructures of the Italian Air Force Meteorological Service (IAF-MS) Observatory. This station is the 34th global station of GAW and aims to characterize the aerosol properties and the processes that influence the troposphere background conditions. In particular, continuous measurements of particle concentration in the accumulation and coarse mode (since August 2002) and Black Carbon (BC) concentration (since July 2005) have been conducted at this measurement site (Marinoni et al., 2008). The other three network points, in Forni, Dosdè (in central Alps) and Gigante glaciers (at Mount Bianco) are continuously measuring standard weather

TABLE 2.1: SHARE Network - AMS (Atmospheric monitoring station), AWS (Automatic weather stations)

Installation site	Country	Type	Altitude
Mt. Cimone (Northern Appennines)	Italy	AMS	2165m
Forni glacier (Central Alps, Valtellina)	Italy	AWS	2669m
Dosdè Glacier (Central Alps, Valtellina)	Italy	AWS	2740m
Gigante Glacier (Mt. Bianco, Alps)	Italy	AWS	3500m
Pyramid Laboratory Observatory (Lobuche, Khumbu Valley)	Nepal	AMS	5079m
Pyramid Laboratory Observatory (Lobuche, Khumbu Valley)	Nepal	AWS	5050m
Pheriche (Khumbu Valley)	Nepal	AWS	4258m
Namche Bazaar (Sagarmatha National Park Head Quarter, Khumbu Valley)	Nepal	AWS	3560m
Lukla (Khumbu Valley)	Nepal	AWS	2660m
Changri Nup (Changri Nup Glacier)	Nepal	AWS	5700m
Kala Patthar (Khumbu Valley)	Nepal	AWS	5600m
Mt Everest South Col	Nepal	AWS	8000m
Urdukas (Baltoro glacier, Baltistan)	Pakistan	AWS	3926m
Askole (Baltistan, Pakistan)	Pakistan	AWS	3015m
Mt. Rwenzori (Elena Glacier)	Uganda	AWS	4700m

parameters such as: air temperature, humidity, wind speed and direction, and solar radiation. Forni is the largest Italian valley glacier (12.5 km2 of surface area) and the AWS analyzes the glacier micro- climate by collecting surface meteorological data (Citterio et al., 2006). Dosdè is the second Italian permanent station and was installed on August 14, 2007. It is located at 2850 m a.s.l.. It permits to collect data on glacial thermal conditions and incoming/outcoming energetic fluxes. This automatic weather station is the highest Lombardy permanent station on glacier and its data are comparable to those collected by AWS Forni in order to verify the effects of climate change on glacial size and micro- meteorological parameters. The larger number of AWS installed in the middle of Himalaya region are present in Nepal. In fact, Himalaya is very sensitive to climate change due to the high variation in altitudes and Nepal is ranked as the fourth most vulnerable country in the world based on 2010 vulnerability assessment and mapping of climate change vulnerable countries ((Siddiqui et al., 2012), (Bonasoni et al., 2010a), (Ueno et al., 1996), (Tartari et al., 1998), (Bertolani et al., 2000), (Ueno et al., 2001), (Bollasina et al., 2002), (Ueno and Pokhrel, 2002)). The Nepal network allows to have meteorological and atmospheric measurements varying in altitude from about 2500 m to 8000 m above the sea level. In particular automatic weather station has been installed in Khumbu Valley, in Changri Nup Glacier and in the Everest South Col to measure meteorological parameters (see Appendix A Fig. A.1 and Fig A.2).

Particularly important is the Nepal Climate Observatory - Pyramid composed by an

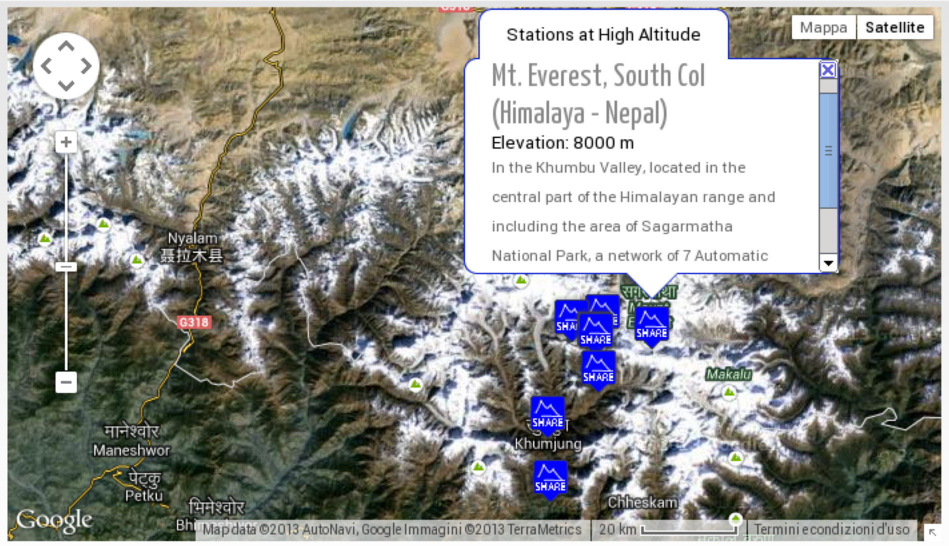


FIGURE 2.2: Nepal monitoring network. Map extracted from SHARE Geonetwork

atmospheric monitor station (ABC-Pyramid at 5079 m a.s.l) and an automatic weather station (5050 m a.s.l.). These installations were deemed to be essential to obtain information on the atmospheric background conditions of this region (Bollasina et al., 2002). The atmospheric monitor station has been equipped to perform continuous measurements of chemical (organic and inorganic, soluble and insoluble), physical (mass and number size distribution) and optical (absorption and scattering coefficients) properties of aerosol. Surface ozone and climate altering halocarbon concentrations are also measured at ABC-Pyramid. Within the AERONET program Aerosol sun photometry studies are carried out as well (Bonasoniand et al., 2007). The so called Pakistan Karakorum Network includes two stations, Askole and Urdukas, and has been installed with the objective to obtain, by measurements of atmospheric and hydrological parameters, a complete meteo-climatic characterization of the Upper Indus Basin and the understanding of the interaction between western weather patterns and monsoon circulation in the Northern Pakistan region (Bonasoni et al., 2010b).

2.2.2 Data Validation

The meteorological data collected were validated in two different ways, from 2002 to 2009 the validation followed the guidelines defined within the CEOP criteria (Coordinated Energy and Water Observation Project) (CEO), whereas, from 2010 to 2011, the validation process took place under WMO recommendations on quality control and data validations, as per WMO reference document (Zahumensky, 2004). The CEOP criteria are based on a visual check, looking for extremely and unusual low/high values and/or periods with constant values. Nocturnal radiation data have been checked for non-zero values, wind speed and direction for sensor freezing and/or unusual high values. Where possible, crosschecking among the variation of different measured parameters (e.g., precipitation with relative humidity) was also performed to assure the consistency among the variations of different variables under the same conditions. The quality control flags follow the CEOP data flag definition document and defines the data as "bad", "dubious", "good", "inconsistent" and "interpolated" Dew Point Temperature, Specific Humidity and U and V Wind components and are computed using "CEOP Derived Parameter Equations". Data collected from 2009 up to now are validated following the WMO recommendation (Zahumensky, 2004). WMO defines the data recorded by the AWS loggers in their original format according to the original sampling interval and before any averaging as Raw Data. For the SHARE-AWS, raw data storage is managed by the AWS manufacturer, and any direct intervention on the acquisition procedures is related to manufacturer interventions. For most of the SHARE-AWS, due to memory storage and data transmission limitations, only automatically processed data (averaged, cumulated or instantaneous values on 1-hour or 30-minutes basis) are stored by data loggers. Processed data are automatically calculated by applying gross automatic data validity checking (basic quality control procedures) on original raw data. Automated quality checks are based on rough plausibility checks designed to remove erroneous sensor information (i.e. recorded values should not exceed the instrumental range). Processed data represent the SHARE- AWS Level-0 data and are defined as data that derive from valid raw data reduction or averaging. According to this definition, for the SHARE-AWS, processed data are represented by the Level-0 data, i.e. 1-hour or 30-minute averaged/cumulated/instantaneous data which are calculated by the data-logger from valid raw data. Quality control and validation procedures on processed data are undergone at the Ev-K2-CNR Data Processing Center, internal consistency (plausible value check, time consistency check) and inter-variable consistency are checked. After quality checks and visual inspection by operators at the Ev-K2-CNR Data Processing Center, processed data are reduced to the Level-1 (data with quality flags : "bad", "dubious", "good", "inconsistent") and Level-2 (only data flagged as "Good"). In the validation of rain data there are problems in terms of continuity, and in that case, the null value's validation criterion (essentially the choice of flag Good or Dubious) of precipitation is different. Modification in validation process are in progress to overcome this gap.

2.3 Database Implementation

As mentioned above, data acquired from high elevation sites allow researchers to improve their understanding of how mountain climate affect global climate. Following this challenge, Ev-K2-CNR Committee has built up a data network that is currently unique, as certified in the partnership with international research programs (see Introduction) and the increasing interest of researchers. The huge quantity of data collected by the Ev-K2-CNR Committee needs to be ordered, stored and maintained in a coherent form. To handle this task, we have constructed a data information system built around a database management system (DBMS). In the future this information system will be extended to collect data also from other data providers (not only from the Ev-K2-CNR Committee) to further facilitate the in depth study of mountain ecosystems in general. The project required a database system that was efficient, proven, standard-complaints, easy to maintain, adaptable and based on open source principles. Consequently, the project oriented toward the use of the Weather and Water Database (WDB) system, developed by the Norwegian Meteorological Institute (T.N.M., 2012). WDB is a database system designed to store meteorological, hydrological and oceanographic data. The system is capable of handling field data (meteorological forecasts and analysis, oceanographic wave and circulation models) as well as observations and point forecasts. WDB is an open source system, its core is a list of SQL function, that built the database schema and form the Command Interface (WCI). WDB is developed using PostgreSQL extended with PostGIS, and other open source components: GRIB API, and GNU tools (e.g. g++, libtool, etc.). It runs on Linux and is released under GPL (GNU General Public License). The license means that it may be used, modified, and distributed by anyone free of charge for any purpose, be it private, commercial or academic. WDB is a proven and reliable system, and it has been a critical component in the Norwegian Meteorological Institute's information infrastructure since 2009. Among other uses at the institute, it is the central data storage component in the yr.no weather service, the most popular Scandinavian weather service with more than five million unique users each week (as of 2012). The system remains in active development, and continues to be modified to make it easier to use, deploy and maintain and to adapt it to new usage areas within meteorology. The main features of the WDB system are: to support high availability production services (as a data storage solution for real-time and archive data)

- to support most types of meteorological, hydrological, and oceanographic data
- to provide a flexible system that can easily be extended with new data types and data formats
- to be easy (and cheap) to maintain and operate
- to provide a simple and consistent interface to all the different kinds of data

2.3.1 Material and Method

WDB version 1.2.0 (currently 1.5.0 is released) was installed on a Linux machine running Debian version 6.0 (i386) (codename “squeeze”). Some adjustments had to be made in the installation process, as the documentation was found to be not always up to date. There were also a number of issues caused by version discrepancies between the libraries used in Debian and Ubuntu (the system used at the Norwegian Meteorological Institute). To support the execution of SQL statement pgAdmin3 has been installed, an open source program to administrate and to develop platform for PostgreSQL. The WDB system itself offers an API (the WDB Call Interface or WCI), based on function calls executed through SQL, that can be used to both read and write data to the database. The benefit of this layer of abstraction is that it allows the underlying tables and views to be modified easily, without requiring changes in the client programs. The core of WDB is a list of 36 SQL functions to set up the entire schema (schemaDefinition.sql) and functionality of the database system (administration, geometry, parameters, etc.). The “schema definition” initializes mainly:

- WDB_SCHEMA (usually wdb_int) is the core schema of the WDB system. It contains the core data tables, functions, and views;
- WCI_SCHEMA (usually wci_int) is a schema that contains the internal functions, views, and tables utilized by the WCI; WCI is the schema that contains the external functions of the WDB Call Interface;
- test is a schema that contains views, functions and tables that are specific for the testing of the WDB system.

Data from AWS, whether raw or validated, must be loaded in the database whenever it is necessary, and made it available to the researchers community via web queries. The process from data harvesting to web queries is sketched in Figure 2.3.

Before the loading process to the database, two operations are required: 1) database adaptation; 2) data pre-processing. The first operation includes WDB settings and parameters addition according to physical quantity and elaborations from AWS. The second operation is writing or coding data from AWS in a compatible way with the WDB metadata scheme for point data. Finally, a PHP page is the core of the connection between the database and the web queries.

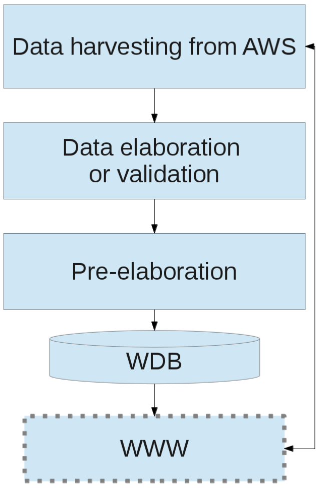


FIGURE 2.3: Data flow: from harvesting to web queries

2.3.2 WDB adaptation

WDB is a general database for storage weather/water data and at the time this project started, was primarily optimized for storing gridded data (version 1.5 added significant new functionality and optimizations directed at point data storage). In this section, we discuss the adaptations and customizations carried out in the Share Geonetwork project to adapt and customize WDB for storing point data from AWSs at high altitude. A data item in WDB is either a number or a grid, and could be either an observation, a forecast, or an analysis. Each data value is uniquely identified and described by a set of dimensions. These dimensions are: data provider; place; reference time; valid time; value parameter; level; data version (Table 2.2). These dimensions are the entry point into the database, and are used for the purpose of searching for and retrieving data. Before we explain the dimensions, it is necessary to understand the concept of a namespace, as it is used in the WDB system. WDB defines three namespaces: a data provider namespace, a place namespace, and a parameter namespace. The namespace is essentially a mapping from a set of names to a set of metadata. This allows for a great deal of flexibility in handling the complexities of real-life metadata. For instance, a meteorological station may be identified by several different indicators or index numbers; each of these indicators may be defined within their own namespace. Similarly, a place will often have different names in different languages; namespaces allow each language to be separated out so that one can have one namespace containing the English names, and another containing Italian names. The only restriction, in this context, is that each metadata item is unique within a namespace in order to facilitate effective search and data retrieval.

Table 2.2: Data dimensions

Dimension	Description
Data Provider	identifies the source of the data; or literally, the entity that provides the data
Place (Geographic Location)	the position of the item on the Earth in a 2d space. In WDB , the geographic location is specified using longitude and latitude in a WGS84 coordinate system by default (though this can be changed when database is set up). It is possible to search for data using coordinates, as well as place names defined in the metadata referring to coordinates. Searches can be prefixed with a spatial interpolation option such as 'nearest' or 'bilinear'
Reference Time	is the moment when the data item is referenced from. For forecast data, this would typically be the reference time of the data values the forecast is based upon; for observation data, it would typically be the time when the observation data was recorded
Valid Time	is the time period for which the data item is valid. The valid time is always stored in the database as a time interval (in the case of a time point, the interval would simply have a distance of 0)
Value Parameter	each data value can be described using a "value parameter". The value parameter is a name that describes the physical or code table basis of the parameter value. The value parameter concept in WCI is broadly similar to the concept of meteorological parameter used in, e.g., GRIB files and is based on the CF metadata standard
Level	is normally used to designate the altitude or depth of the data value. Level is designated using a level interval (height from and to) and a level parameter (e.g., height above sea level, pressure)
Data Version	there can be several different versions of the same data value that is valid for the same time, position, etc. This can happen with probability forecast calculations or when a data value is edited

Namespaces are identified using a numerical ID. The default namespace ID is 0, which contains metadata automatically generated from the base metadata in the database. The namespaces from 1 to 254 are reserved for usage by WMO centers; all other numbers can be freely used to define project/institute-specific namespaces. Before any other WCI function call, wci.begin("user") initializes the WDB Call Interface for a specific user, with the users default namespace. The default installation of WDB contains only a limited subset of basic metadata; for instance, it contains only 62 of the most common meteorological parameters. Thus, it is usually recommended to define a new namespace and add metadata as required by the application. It is possible to define a new namespace using the WCI function: wci.addNamespace(). The next script (Fig. 2.4) is used to configure the WDB system setting the new namespace, the data provider and organization, new persons involved in the use of the database and adding new high mountain sites.

```
-- WDB inizialization, with the default Namespace
SELECT wci.begin('wdb');
-- WDB users adding
SELECT wci.addPerson('Filippo','Locci','Dr','Hi Filippo','FiL',NULL, 'male',
                     NULL, NULL,'2012-09-01','2999-12-31',NULL);
SELECT wci.addPerson('Maria Teresa','Melis','Dr','Hi Teresa','TeR',NULL, 'female',
                     NULL, NULL,'2012-09-01','2999-12-31',NULL);
SELECT wci.addPerson('Francesco','Dessì','Dr','Hi Francesco','FrA',NULL, 'male',
                     NULL, NULL,'2012-09-01','2999-12-31',NULL);
-- Add Organizations section
SELECT wci.addOrganization('Ev-K2-CNR Committee','Ev-K2-CNR','non-profit associattion',
                           '1992-01-01','2999-12-31',
                           'The Committee promotes scientific and technological research in mountain areas');
-- Add new Namespace for Raw data, Namespaces and Organization are related ('Ev-K2-CNR')
SELECT wci.addNamespace(1001,'Raw data namespace','namespace for raw data from AWS SHARE','production',
                        'Ev-K2-CNR','FiL','2012-05-01');
-- WDB inizialization, with the Namespace '1001'
SELECT wci.begin('wdb',1001,1001,1001);
-- Copy default parameters from default Namespace to '1001' Namespace
SELECT wci.copyParameterNameSpace( 0 );
```

FIGURE 2.4: WDB setting script: new namespace, organization and new persons adding

The next step to set up the system giving whole metadata information is the addition of the new data-providers and station points (see Fig. 2.5).

In this project the field dataProviderName in the function wci.adddataprovider() is set to be 'Ev-k2-CNR Committee'. Each dataProviderName is the unique ID of the data provider within its namespace. A data provider is qualified by its type (e.g., computer system, names observation site, person, ship) and domain delivery (point, grid, or any). The latter information helps the database to optimize queries. Before data can be loaded, the database needs to be seeded with the appropriate value parameters. Using the wci.addparameter() function, we have added about 600 new parameters in order to fit with the search and retrieval patterns required by our data retrieval service (see Fig. 2.6). Once the system has been set up with appropriate metadata, it is possible to load data into the database system.

2.3.3 Data loading

The function wci.write() is used to load data: it executes an SQL statement and loads data point in the database (see Fig. 2.7).

Before starting any loading operation the verification that all the metadata information are properly set in the database is required. "data provider", and "placename" has been set in adaptation step, while "value", "referencetime", "validtimefrom", "validtimeto", "levelparameter", "levelfrom" and "levelto" are picked up directly from AWS files. Using a script developed at met.no (wdb-fastload utility) it is possible to load a large amount of data, that takes a text data format from standard input and copies it into the database.

```
-- WDB inizialization, with the Namespace '1001'
SELECT wci.begin('wdb',1001,1001,1001);
-- The new data provider is 'Ev-K2-CNR Committee'
select wci.adddataprovider('Ev-K2-CNR Committee','wci user','Point',
                          '1000 years','Data from Ev-K2-CNR Committee
                           in the framework of SHARE project');
-- For each station the name and spatial coordinate are reqired
-- Add the AWSs in WGS84 projection
SELECT wci.addPlacePoint('Nepal Climate Observatory - Pyramid SHARE',
                         ST_GeomFromText('Point(86.813333 27.959167)', 4030));
SELECT wci.addPlacePoint('Mt. Cimone SHARE',
                        ST_GeomFromText('Point(10.7 44.2)', 4030));
SELECT wci.addPlacePoint('Forni Glacier SHARE',
                        ST_GeomFromText('Point(10.59 46.399)', 4030));
SELECT wci.addPlacePoint('Dosde Glacier SHARE',
                        ST_GeomFromText('Point(10.22 46.39)', 4030));
SELECT wci.addPlacePoint('Gigante Glacier SHARE',
                        ST_GeomFromText('Point(6.93 45.85)', 4030));
SELECT wci.addPlacePoint('Pheriche SHARE',
                        ST_GeomFromText('Point(86.82 27.89)', 4030));
SELECT wci.addPlacePoint('Namche Bazar SHARE',
                        ST_GeomFromText('Point(86.71 27.8)', 4030));
SELECT wci.addPlacePoint('Lukla SHARE',
                        ST_GeomFromText('Point(86.72 27.69)', 4030));
SELECT wci.addPlacePoint('Kala Patthar SHARE',
                        ST_GeomFromText('Point(86.83 27.99)', 4030));
SELECT wci.addPlacePoint('Mt. Everest South Col SHARE',
                        ST_GeomFromText('Point(86.93  27.97)', 4030));
SELECT wci.addPlacePoint('Urdukas Baltoro Glacier SHARE',
                        ST_GeomFromText('Point(76.29  35.73)', 4030));
SELECT wci.addPlacePoint('Askole SHARE',
                        ST_GeomFromText('Point(75.81 35.68)', 4030));
SELECT wci.addPlacePoint('Mt. Rwenzori Elena Glacier SHARE',
                        ST_GeomFromText('Point(29.91 0.39)', 4030));
SELECT wci.addPlacePoint('Pyramid Laboratory Observatory SHARE',
                        ST_GeomFromText('Point(86.81 27.96)', 4030));
SELECT wci.addPlacePoint('Changri Nup Glacier SHARE',
                        ST_GeomFromText('Point(86.7647 27.9822)', 4030));
```

FIGURE 2.5: WDB setting script: data provider and new high mountain sites adding

```
-- WDB inizialization, with the default Namespace
SELECT wci.begin('wdb',1001,1001,1001);
-- Adding hourly temperature parameters
SELECT wci.addparameter('hourly min air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('hourly max air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('hourly average air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('hourly std deviation air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('min air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('max air temperature',null,null,null,null,null,null,'Cel');
SELECT wci.addparameter('std deviation air temperature',null,null,null,null,null,null,'Cel');
```

FIGURE 2.6: WDB setting script: new parameters adding

```
SELECT wci.write
(
        'value'          -- gid,
        'placename'      -- text,
        'referencetime'  -- timestamp without time zone,
        'validtimerom'   -- timestamp without time zone,
        'validtimeto'    -- timestamp without time zone
        'valueparameter' -- text,
        'levelparameter' -- text,
        'levelFrom'      -- float,
        'levelTo'        --float
)
```

FIGURE 2.7: WDB loading script: wci.write()

```
def generate_data():
    # assign value, place, referencetime, etc..
    yield value, place, referencetime, validfrom, validto, param, level, level_from, level_to

def writeit(dataprovider,count,out_file):
       if count==1:
              out_file.write(dataprovider + '\n') # optionally add namespace identifier
               for data in generate_data():
           out_file.write(str(data[0]))
           for element in data[1:]:
               out_file.write('\t')
               out_file.write(str(element))
           out_file.write('\n')
```

FIGURE 2.8: Python conversion format script

Data from the stations, raw or validated, have a well defined format, hence before starting loading data pre-processing is required. The pre-processing core is a Python script that takes the raw or validated file as input and converts it in a wdb-fastload format (see Fig 2.8).

In order to make the loading operations simple and fast, we have developed a Python GUI using the program Glade (see Fig. 2.9). With this loading tool is possible adding new data provider in the database (only for administrator users), convert files in wdb-fastload format (now for fifteen data format corresponding to the AWS data logger) and then load the selected and converted file in the database. Via this graphical tool it is possible also for unskilled users to load AWS data without having to launch command in a text shell (see Fig.2.9).

2.4 Web queries from SHARE Geonetwork website

Data access is now possible with the new version of SHARE Geonetwork platform (published in November 2013) visiting the address http://www.geonetwork.evk2cnr.org/index.php (see Fig. 2.10).

In the section: "SEARCH IN HIGH ALTITUDE AND ENVIRONMENTAL DATA", choosing the radio button "dataset", the user can query the High-altitude dataset, both raw (only for authorized users) and validated data. Validated data are quickly available after a registration process, by clicking on "Register now!" (see Fig. 2.11). The .csv is the unique data download format at the moment. In the future netCDF data format will be implemented for download.

After login, the user will be capable to download validated data after a simple query process (see Fig. 2.12). The query process is quite simple, it comprises four selection steps: the data provider selection, the AWS selection, the date and the interval of time

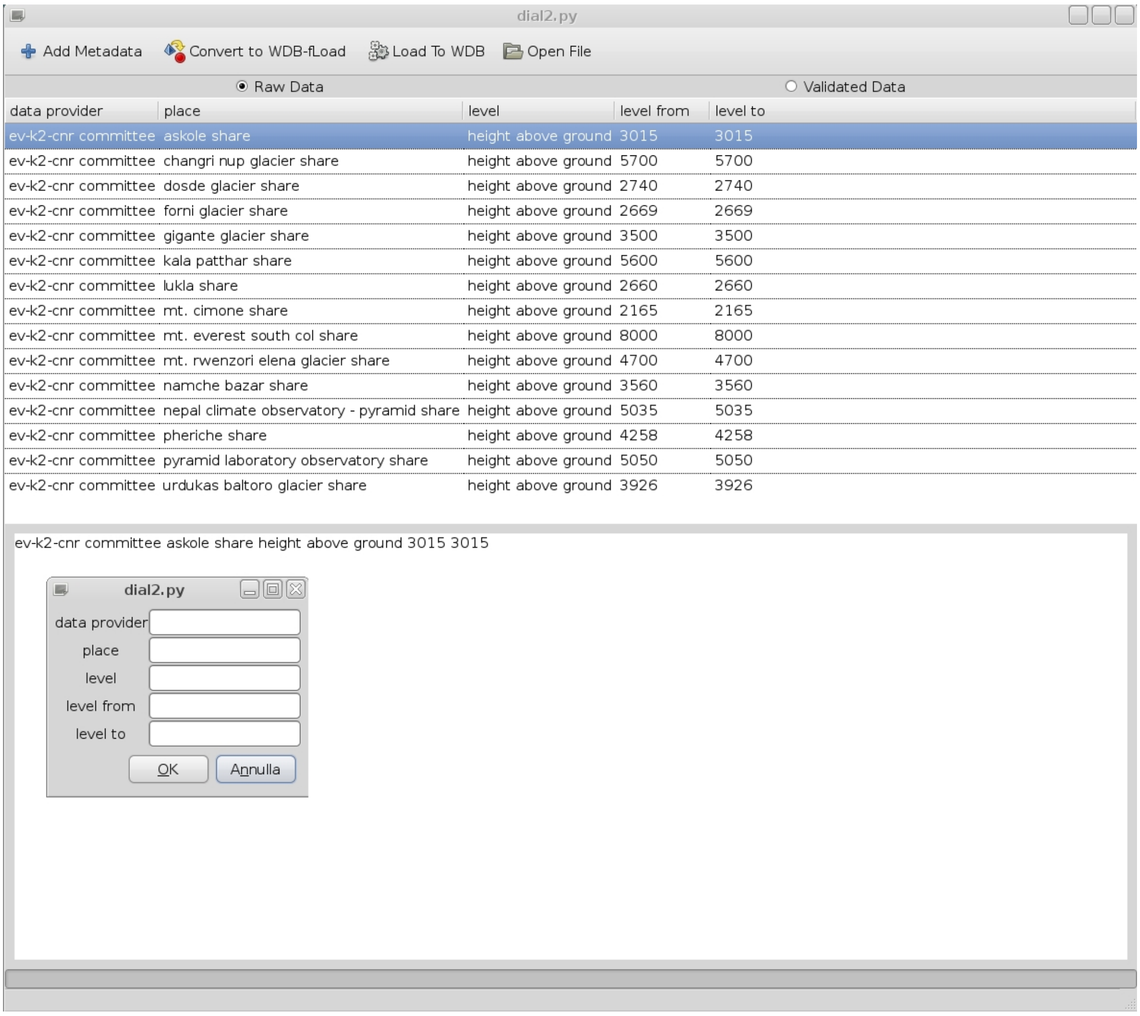


FIGURE 2.9: Python GUI for loading data and metadata into the database

selection and the physical parameter selection. The core of this page is the connection with WDB and the use of the function wci.read() (see Fig. 2.13). Pressing the search button icon the user obtain the result in a tabular or graphical form, if "Table" or "Graph" button are pressed (mutually exclusive radio buttons) see Fig. 2.14 and Fig. 2.15.

Raw data are available after formal request to the EvK2 CNR Committee via "Contact us" section, in the web-site header, or using the "data request form " (see Fig. 2.16).

FIGURE 2.10: Homepage SHARE Geonetwork website

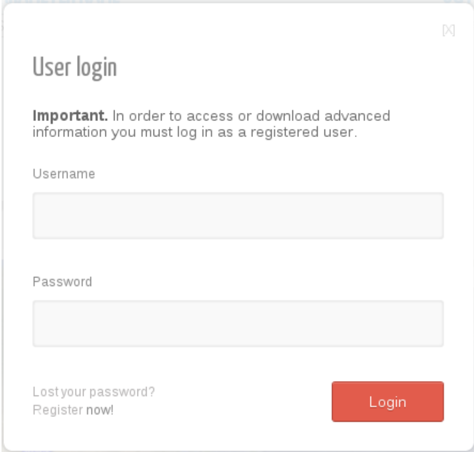


FIGURE 2.11: Login as registered member form.

FIGURE 2.12: Query form example.

```
// attempt a connection
$dbh = pg_connect("host=localhost port=5432 dbname=wdb user=root password=        options='--client_encoding=UTF8' connect_timeout=5");
if (!$dbh) {
    die("Error in connection: " . pg_last_error());
 }

// execute query
$begin = "SELECT wci.begin('wdb',1001,1001,1001)";

$sql = "SELECT * FROM wci.read ( ARRAY['$l'],'$m','inside $yf-$mf-$df 00:00:00Z TO $yt-$mt-$dt 00:00:00Z',NULL,ARRAY['$vp'],NULL,NULL,NULL::wci.returnfloat)";
$result0 = pg_query($dbh, $begin);
$result = pg_query($dbh, $sql);
if (!$result) {
    die("Error in SQL query: " . pg_last_error());
}
```

FIGURE 2.13: PHP connection script to WDB and the use of the function wci.read().

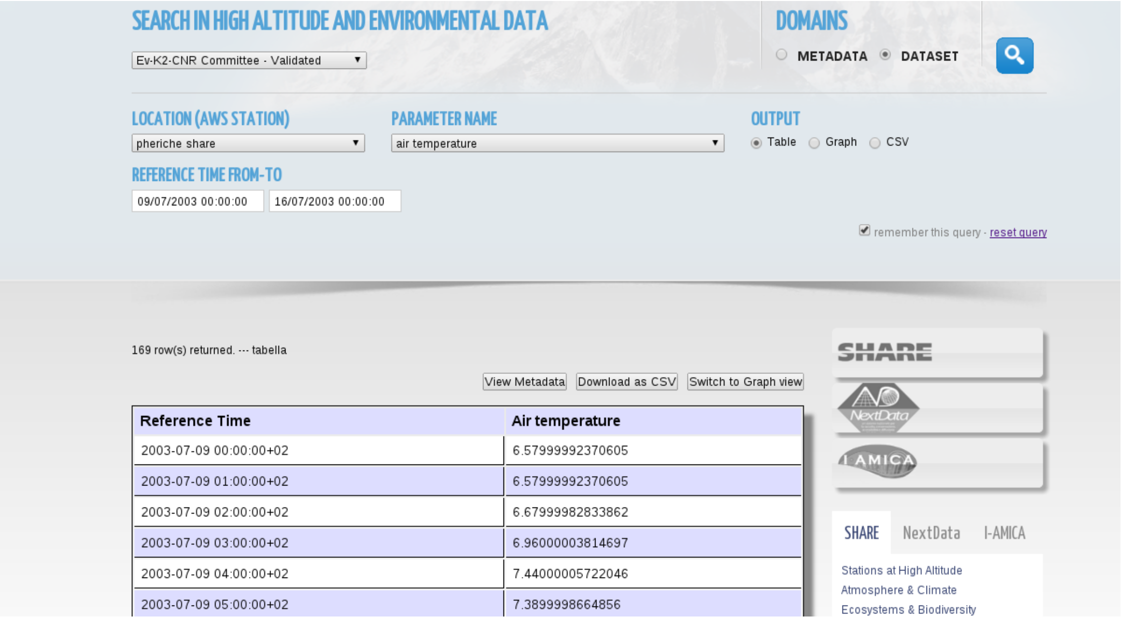


Figure 2.14: Tabular query result.

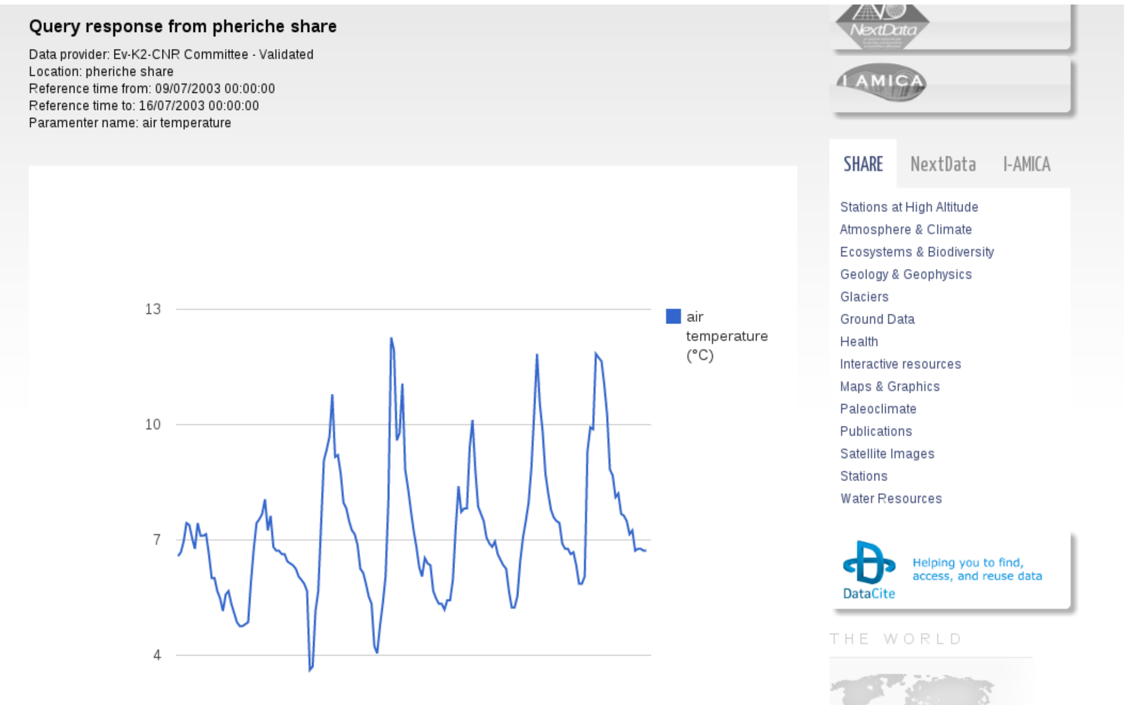


Figure 2.15: Graphical query result.

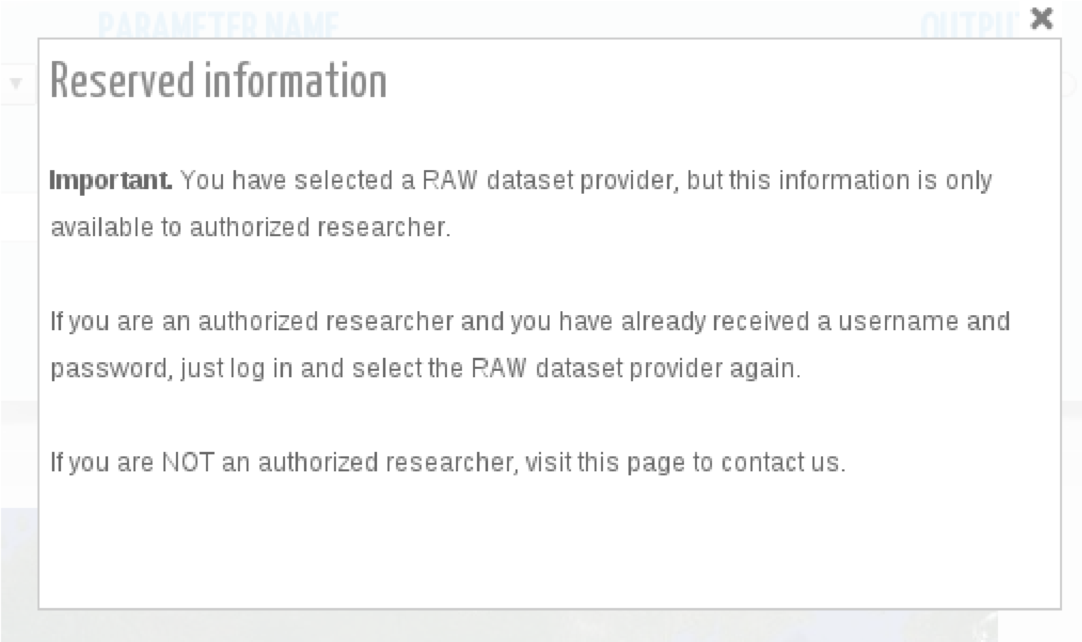


FIGURE 2.16: Reserved access area warning.

Chapter 3

The paleoclimatology database: WDBPALEO

3.1 Introduction

WDBPALEO has been designed starting from the Meteorological, Hydroloical and Oceanografic (MHO) open source database WDB (T.N.M., 2012) to store paleoclimatology data derived from chemical and physical characterization of ice cores and marine sediment cores. The adaptation of an MHO database for ice/sediment core data it has been achievable treating a single core as an observation point (such as an AWS) (Fig 3.1), in fact:

- both cores and weather stations are geographically represented with a couple of coordinates that in a GIS environment can be treated as a geometrical vector point;

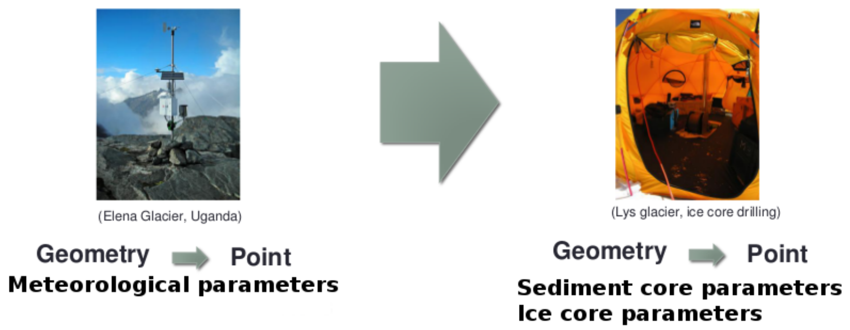


FIGURE 3.1: A single core is assumed like an observation point

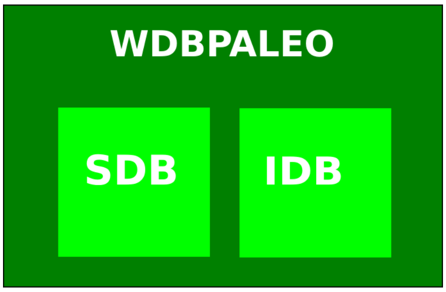


FIGURE 3.2: The WDBPALEO system structure

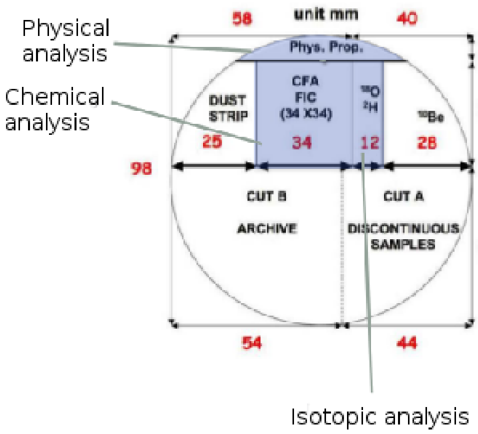


FIGURE 3.3: Sample ice core analysis

- they both record climate data, in particular a core gives information about the past climate trend and the weather station gives information about the current climatic system. These two entity store the same type of data, characterized by a numerical value with a parameter related to a temporal information.

The WDBPALEO system consist of two separated database: IDB (Ice-core Data Base, also WDBPALEO-IDB) and SDB (Sea-core Data Base, also WDBPALEO-SDB) (Fig. 3.2). Each of them are properly adapted to store ice core data or marine sediment core data.

WDBPALEO is a geographical database and the spatial information is defined by a couple of georeferenced coordinates that identify each ice/sediment core and the related chemical/physical parameter, derived from sample analysis (Fig. 3.3). Every single ice/sediment core is stored in the database in longitude-latitude with WGS84 datum (EPSG 4326/4030).

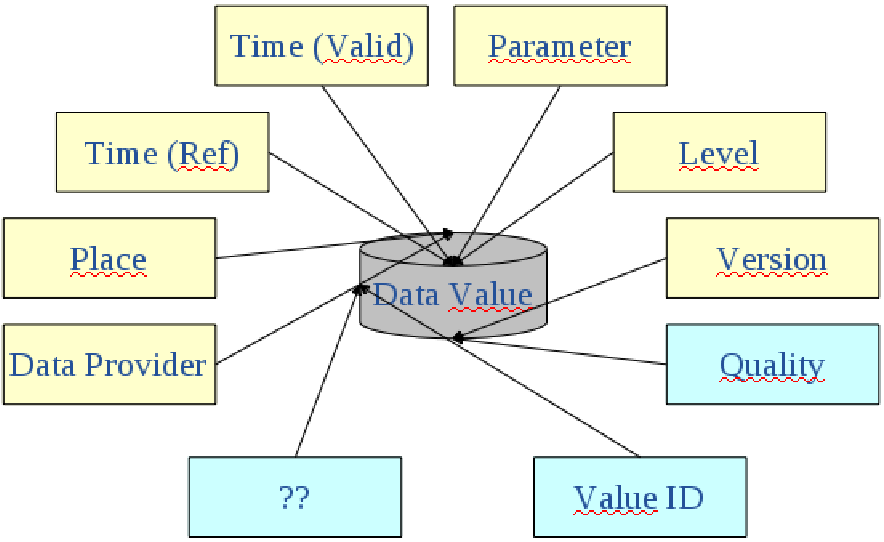


FIGURE 3.4: The WDB system data model

3.2 Main features of the WDBPALEO system

WDB is designed to archive MHO data, but the data model can be adapted to stock data derived from analysis about ice/sediment cores.

Each element of the WDB data model can be related to the WDBPALEO system metadata. In fact the list of elements (Data Provider, Place, Time, Parameter, Level, Version, Data value) is exhaustive to represent uniquely each data derived from ice/sediment core analysis.

The tables where the numeric value about ice/sediment core parameters was chosen to be inserted in are the central core (wdb_int.floatvaluegroup and the related table wdb_int.floatvalueitem) (Fig. 3.5). These two tables (VALUES) are related (relationship) to the other tables represented by the categories PARAMETERS, PLACEPOINT and DATA PROVIDERS which compose the global schema of the database (Fig. 3.6).

In order to fully adapt WDB system for paleoclimate data need to be change the temporal scale of the system itself, properly designed for MHO data. In fact, the first weather observations date back to 1654, so WDB allows to store data that are not older than the XVIIth century.

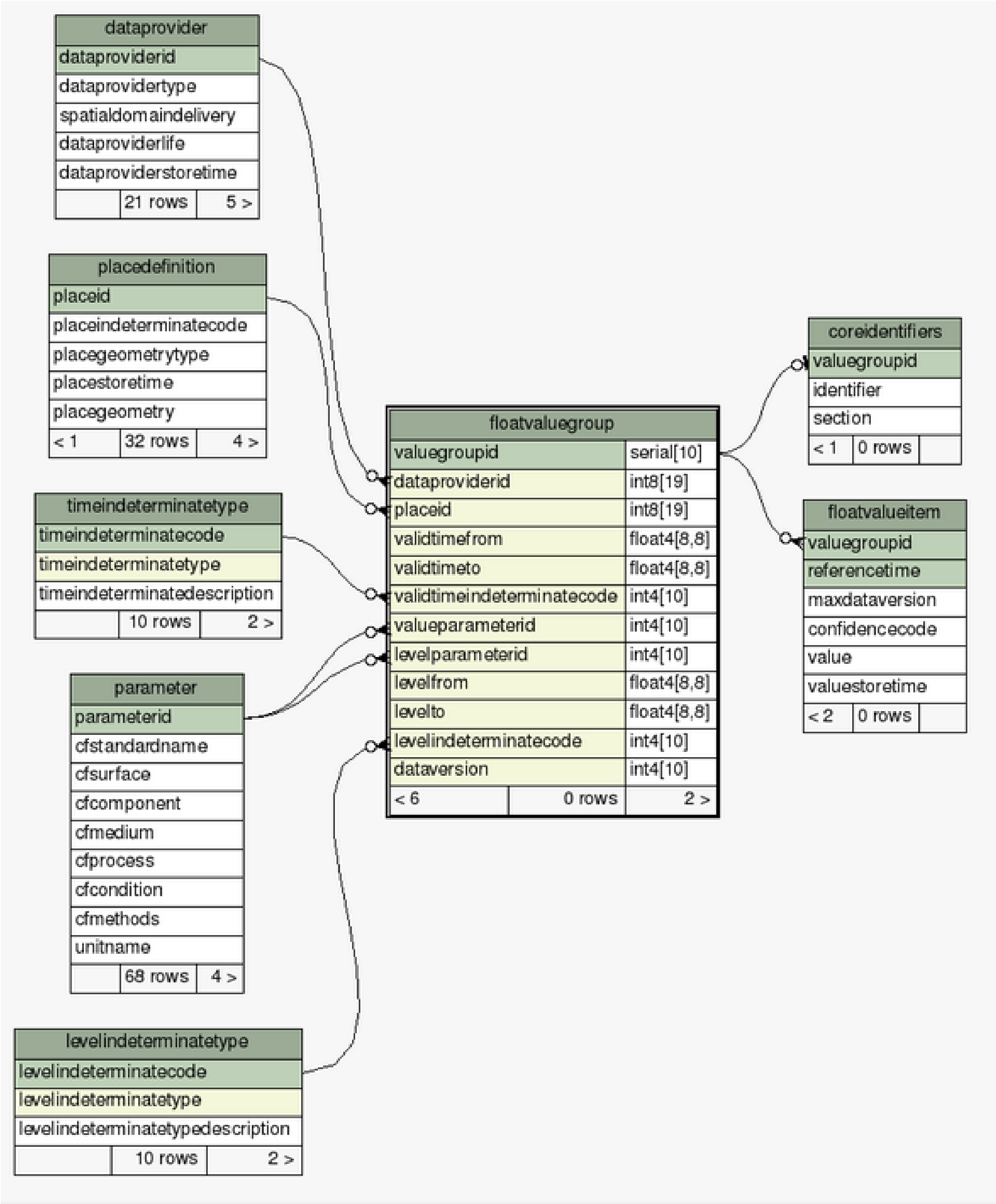


Figure 3.5: WDBPALEO schema sample: the table wdb_int.floatvaluegroup and the related table wdb_int.floatvalueitem to store the parameter's value

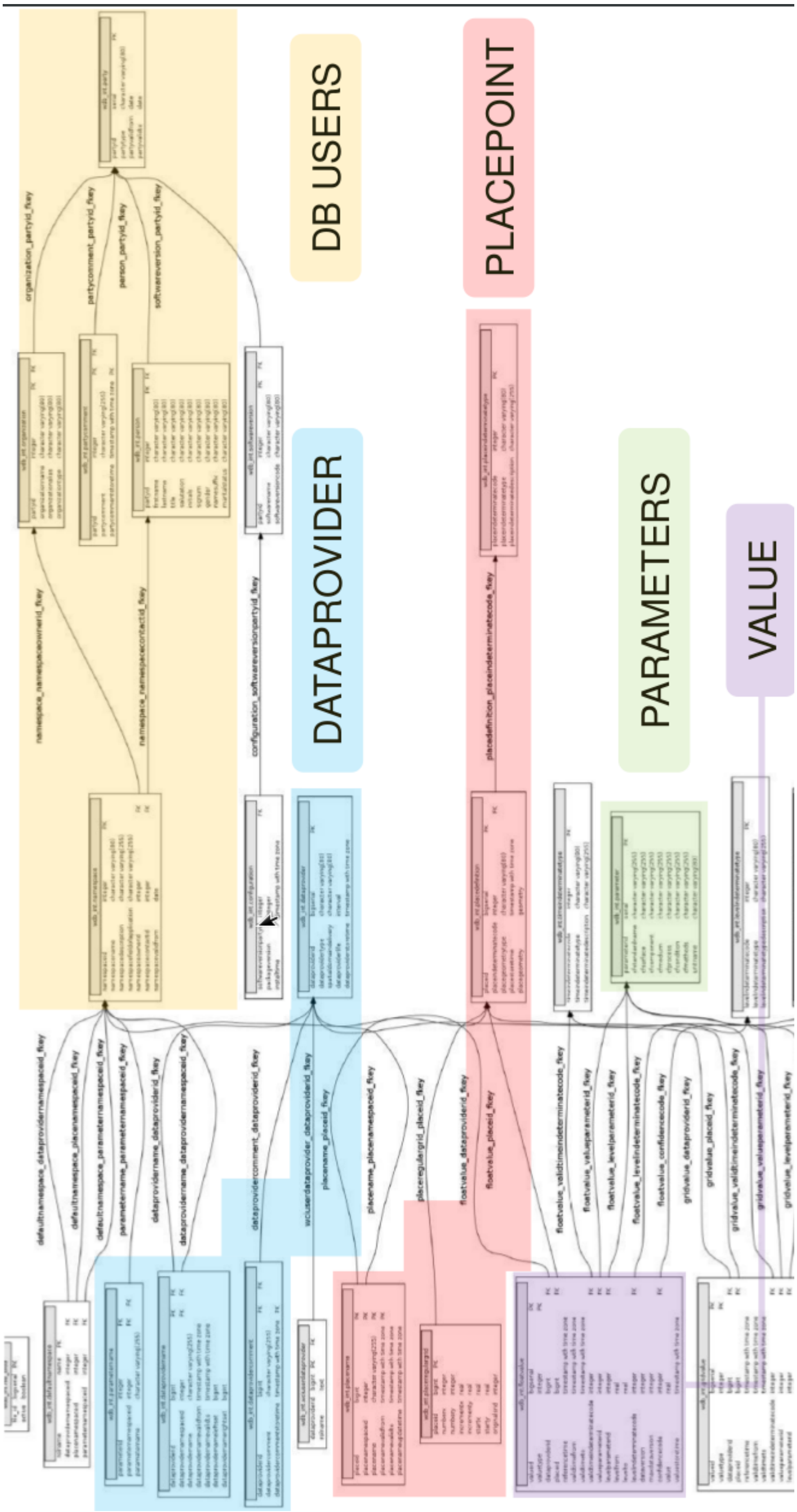


FIGURE 3.6: WDB global schema

3.3 Time field adaptation

The information about time series was fundamental, a lot of problems with the data/time field PostgreSQL have been found. The 'timestamp with time zone' type used to store time field in PostgreSQL has a storage size of 8 bytes. This field type can archive temporal data from 4713 BC to 294276 AD. Inserting data referred to a period before 4713 BC an error occurr. In spite of this, along with WDB wbd-fastload is used to insert raw data in the database. It uses 'timestamp with time zone' type as time field (and Boost library to manage time, with an inferior limit of about 1400 BC). Hence, to load raw data in the database new code have been written. Psycopg Python libraries and Bash scripts have been used for this scope (see Appendix B - Fig. B1, Fig. B2 and Fig. B3).

To solve the problem with PostgreSQL time field we have modified a part of the source code by inserting new functions to write data into WDBPALEO, without time limits. This has increased the value of temporal data aspects that is essential for paleoclimatic analysis. We used a 'real' data type field. Positive numbers are years after Christ and negative numbers are referred to years before Christ. The 'real' field type can archive a 4 bytes information and can be positive, negative and with a precision of six decimals. The WDBPALEO package descend from WDB directly, it has been modified in the source code to accept data without time constraints, without original WDB schema modification. Subsequent modification and adjustment to the WDB schema have been adopted to load ice/sediment core data starting directly from the WDBPALEO package.

3.4 WDBPALEO data loading

In order to load ice core data and marine sediment core data into WDBPALEO two different method and adaptation to the WDB schema and its default parameters have been adopted.

3.4.1 WDBPALEO-IDB characterization

3.4.1.1 Reference system

The predefined reference system (RS) in WDB is the 4030 EPSG. This RS is based on the WGS84 ellipsoid but it has an unknown datum. In a geospatial database the information about geographical position is fundamental. Thus, because ice core drilling is based on morphologic glacier parameters, to offer and to work with spatial correct

```
CREATE TABLE icecorename (
    icecoreid bigint NOT NULL,
    icecorenamespaceid integer NOT NULL,
    icecorename character varying(255) NOT NULL,
    icecorenamedrilling character varying(255) NOT NULL,
    icecorenamevalidfrom timestamp with time zone NOT NULL,
    icecorenamevalidto timestamp with time zone NOT NULL
);
```

FIGURE 3.7: The wdb_int.icecorename table

information the EPSG 4326 (that is used by the GPS satellite navigation system and for NATO military geodetic surveying) has been used.

3.4.1.2 Structures adaptations

To perform a best experience in the data search it was very important to insert a piece of information about the regional area where an ice core was drilled. To archive the zone where the ice cores were drilled a record in the new table wdb_int.icecorename has been created.The wdb_int.icecorename (Fig. 3.7) table has the analogous function of the wdb_int.placename table in WDB

3.4.1.3 WDBPALEO-IDB data loading

The data loading process follows a delicate procedure, based on the ideal concept from which the database is created. According to this, IDB has three main areas where ice core data, dataproviders and parameters are archived. The central core of the database is the table dedicated to raw numerical value (wd_int.floatvalue). A raw numerical value item in IDB can be: an observation, an analysis, etc. Each item consists of a value and a number of dimensions that describe the value. In this table the other three part are called with foreign key to make unique relations between the value and the other parts of database where are archived the information that characterize it (Fig. 3.8).

After running the database setting start the loading data about chemicals and physics characterization. The steps are:

- geographical information setting;
- dataprovider setting;
- parameters adding;

```
CREATE TABLE floatvalue (
    valueid bigint NOT NULL,
    valuetype integer NOT NULL,
    dataproviderid bigint NOT NULL,
    icecoreid bigint NOT NULL,
    referencetime real NOT NULL,
    validtimefrom real NOT NULL,
    validtimeto real NOT NULL,
    validtimeindeterminatecode integer NOT NULL,
    valueparameterid integer NOT NULL,
    levelparameterid integer NOT NULL,
    levelfrom real NOT NULL,
    levelto real NOT NULL,
    levelindeterminatecode integer NOT NULL,
    dataversion integer NOT NULL,
    maxdataversion integer NOT NULL,
    confidencecode integer NOT NULL,
    value real NOT NULL,
    valuestoretime timestamp with time zone NOT NULL
);

ALTER TABLE wdb_int.floatvalue OWNER TO daniele;
```

FIGURE 3.8: The wdb_int.floatvalue table for IDB

- upload data from ice core analysis (raw numeric value).

The first part data loading involve the **geographic information setting**. In most cases, the spatial information obtained from papers reports the identical coordinate for different ice cores drilled in the same glacier. This problem arise from the fact that the GPS coordinates were taken with poor precision and it refer to the drilling site and not to a single ice core. In order to respect the topological rules, to insert the geographic information in to IDB, a GIS operation (shift points) was applied; this function moves overlapped points with same coordinates in a circle around the original position. At the end of this operation, 178 points with ice core name and drilling site attributes were stored (Table 3.1).

Table 3.1: Dataprovider stored in to IDB database.

Ice core name	**Drilling site**	**Reference for ice core spatial position**
bl2001 1	Belukha glacier (Siberian Altai Mountains)	Henderson K, 2006
cdl03/1	Colle del Lys	
cdl96	Colle del Lys	
d-1 dunde	Dunde Ice Cap	Thompson L.G, 1990
d-3 dunde	Dunde Ice Cap	Mosley-Thompson E., 1989
dasuopo c1	Dasuopu Glacier	Thompson L.G, 2000
dasuopo c2	Dasuopu Glacier	Duan K, 2000
dasuopo c3	Dasuopu Glacier	
eclipse icefield icecore 1	St. Elias Mountains	Yalcin k, 2007; 2006; 2003;2002; 2001
eric2002a	East Rongbuk Glacier	Xu, J, 2002; Ming, J, 2008; Hou S, 2007
eric2002c	East Rongbuk Glacier	Xu, J., 2002 and Ming J, 2008
fedchenko c1	Fedschenko Glacier (Pamir Mountains)	Aizen V, 2009
fedchenko c2	Fedschenko Glacier (Pamir Mountains)	Aizen V, 2009
fremont glacier 91-1	Fremont glacier	Schuster P.F. et al., 2002, 2000; Naftz D.L. et al., 2002
fremont glacier 98-4	Fremont glacier	Naftz D.L. et al., 2002
fwg kilimanjaro ice core	Kilimanjaro Furtwangler Glacier	Thompson L.G, 1979
guliya c7	Guliya Ice Cap (Western Kunlun Mountains)	Yang M, 2006; Thompson L. G., 1995, 1997; Yang M, 2000
guoqu c2	Guoqu Glacier	Grigholm B, 2009; Zhang Y, 2007
hsc1 huascaran 1	Col of Nevado Huascaran	Thompson L. G., 1995
hsc2 huascaran 2	Col of Nevado Huascaran	Thompson L.G, 1993

Continued on next page

Table 3.1 – *Continued from previous page*

Ice core name	**Drilling site**	**Reference for ice core spatial position**
inilchek c1	South Inilchek Glacier (Tien Shan Mountains)	Yao T, 2008
lg1 kenya ice core	Kenya Lewis Glacier	Thompson L.G, 1979
lg2 kenya ice core	Kenya Lewis Glacier	Vincent C.E, 1979
mount logan pr col ice core	Mount Logan	NOAA database
nif2 kilimanjaro ice core	Kilimanjaro Northern Ice Field	Thompson L.G, 2002
nif3 kilimanjaro ice core	Kilimanjaro Northern Ice Field	Thompson L.G, 2002
puruogangri c1	Puruogangri Ice Cap	Thompson L.G, 2006
puruogangri c2	Puruogangri Ice Cap	Yao T, 2008
quelccaya core 1	Quelccaya Ice Cap	Zagorodnov V. 2005
quelccaya core 2	Quelccaya Ice Cap	Zagorodnov V. 2005
sc-1	Sajama Ice Cap	Ginot P, 2010
sc-2	Sajama Ice Cap	Ginot P, 2010
sif1 kilimanjaro ice core	Kilimanjaro Southern Ice Field	Thompson L.G, 2002
sif2 kilimanjaro ice core	Kilimanjaro Southern Ice Field	Thompson L.G, 2002
All 133 ice cores		
Malan ice core	Tibetan Platou	Yao T, 2003

Table 3.2: Dataprovider stored in to IDB database

Dataprovider name	**Ice core investigated**
Eichler A.	Bl 2001 1
Yalcin K.	Eclipse Icefield IceCore 1
Ming J.	ERIC 2002C
Thompson L.G.	Dasuopo C1-C2-C3
Aizen V.B.	Fedchenko C1-C2
Shuster P.F.	Fremont 91-1 98-1 98-4
Kaspari S.	ERIC 2002A-C
Grigholm B.	Guoqu C2
Kreutz K.J.	Inilchek C1
Osterberg E.	Mount Logan PR Col Ice Core
Campen R.K.	Sajama SC-1
Maggi v.	Lys

The **Data Provider** identifies the source of the data; literally, the entity that provides the data. It can be the person in charge of ice core drilling or the principal investigator about the ice cores analysis. A data provider is identified by the DataProviderName that is used to search for the data into the database. Finally, in the WDBPALEO-IDB system 12 data providers have been set (Table 3.2).

The WDBPALEO-IDB **Parameter** identifies the characteristic or measurable factor of the value being parameterized. Parameters provide a definitive description of what the data represents, including chemical and physical properties. When ice cores are drilled they are cut and prepared to be cataloged and stored for analysis. As shown in Figure 3.9, the ice core is divided in different sector and each sector is intended for a different type of analysis. Ice cores contain many proxy-parameters useful to scientists for past climate reconstruction. For example, regarding chemical analysis, the concentration of atmospheric trace gases such as nitrous oxide (N2O), methane (CH4) and carbon dioxide (CO2) provide information about natural variation and changes in atmospheric composition due to anthropic actions. Otherwise, the physical analysis supply different information. Conductivity allow to investigate volcanic activity or particle size and concentration give information of wind speed. The ice measurements aim to determine the chronological extension of possible atmospheric records. In particular, non-destructive measurements, such as FTIR (infrared) and DEP (dielectric properties), aim to define horizons of historical reference (137Cs, 3H, volcanic tephra, insoluble powder levels), for determining seasonality and compositional trends (oxygen and hydrogen stable isotopes, ice chemistry, mineral powders contained, and others). After an accurate investigation about the main physical and chemical factors, 80 parameters with a proper measurement unit were selected (Table 3.3). To standardize the data, each parameter was defined by a IUPAC name for chemical value and SI (International System of Units) for units of

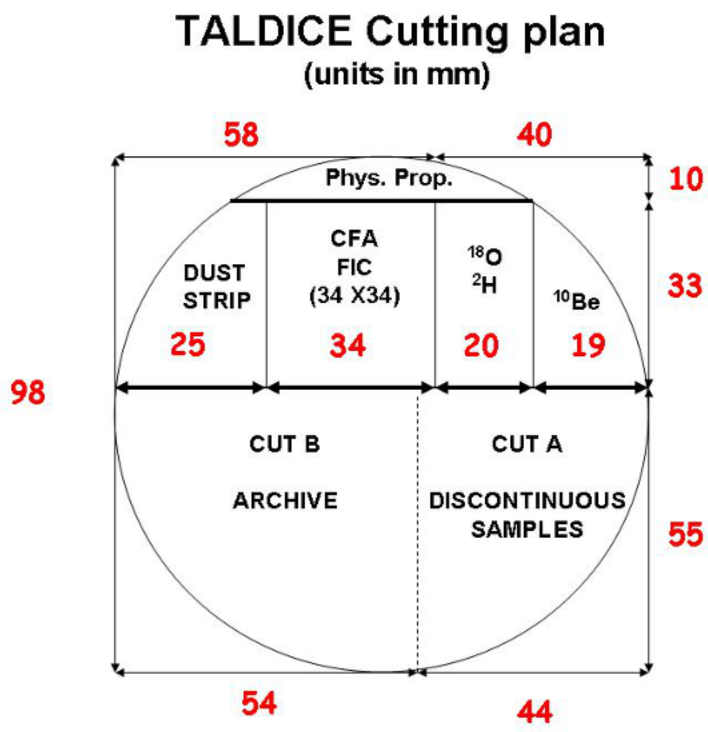


FIGURE 3.9: Taldice cutting plan

TABLE 3.3: List of some parameters and measurements unit in the IDB database.

Parameter name	**Measurement unit**
d18O	Ratio
Calcium	Ppb
Chloride	Ppb
Ammonium	Ueq/L
Cerium	Ppb
Conductivity	uS per cm
Fluoride	ppb
All 72 parameters	
D (CO2/N2)	‰with respect to present

measurements. Afterward, data were stored in the IDB “parametername” table.

Raw numeric value is for the number obtained from a specific chemical or physical analysis on ice core sample. E.g., -14.62 d18O, -0.30 Tanom or 87.6 ng/l Pb. This step was critical as each raw numeric value of chemical and physical measure are linked to the three parties mentioned above (ice core, dataprovider and parameters). First, before loading data in to IDB, the original .txt and .xls files have been modified in .CSV format. After a careful analysis of these requirements, 281.728 record of raw values have been inserted into the database. A bash shell script was developed to automate data upload, even in large quantities (See Appendix B for details, Fig. B3). The bash shell script

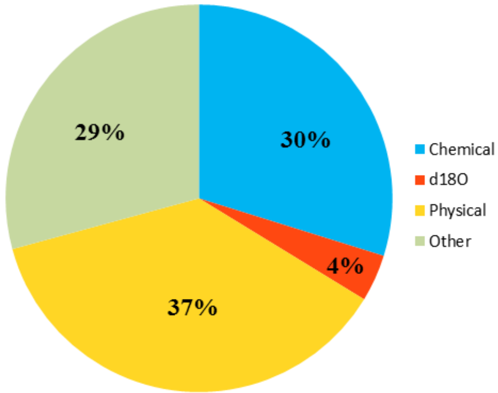


FIGURE 3.10: Percentage and type of data stored

has been deployed in order to simplify the loading procedure. It reads the .CSV file where ice core data were been antecedently prepared with a precise column order and then connecting to the database. After it recall a SQL function (wci.writepaleo) that can distribute each attribute inside the right tables fields. This software have the aim to help future users in loading new data. The SQL function used to write data in the database was built with idea to controlled the data that were going to be insert in the database to avoid redundancy and other errors (see Appendix B, Fig. B4). The analysis about the ice cores were available only for 36 of the 178 that stored in the database. As showed in figure 3.9, the oxygen 18 is one of the most common ice core proxies in the analysis of stable isotopic ratios. In fact, the 4% of the entire value stored in the database are referred to d18O ratio. This data used to reconstruct time-series of past temperature. The 30% of the loaded data is related to the chemical analysis to evaluate the amount of different elements in the ice cores samples. The values are expressed in different measurement units for example mq/l, ppb or ueq/l for the Ammonium, Calcium, Sulfate and the other chemical elements. As regard physical investigation, the values stored in IDB are referred to different analysis, such as conductivity, accumulation rate, layer thickness and represent the 37% of the entire value (Figure 3.10). The table wdb_int.floatvaluegroup is linked to each other through the primary keys (id). For this reason, the loading data in to the database must strictly enforce these ties (Figure 3.10). The user can query the WDBPALEO-IDB database with elaborate requests, crossing data from different tables. As a consequence, it is possible to consider the same object at the same time from different points of view (e.g. dataprovider, ice core, drilling site, parameter, etc.) (Sumathi and Esakkirajan, 2007).

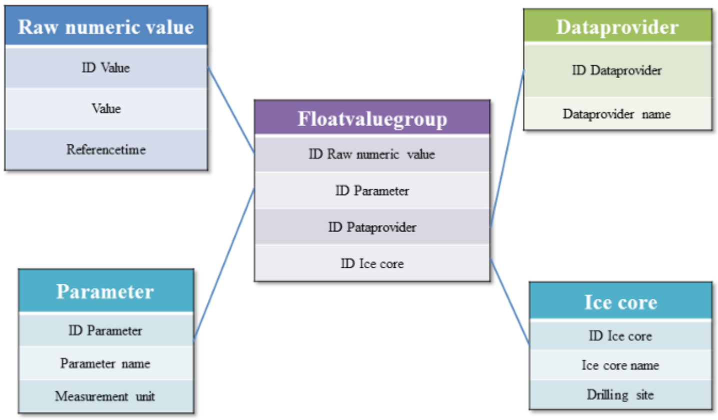


FIGURE 3.11: WDBPALEO-IDB ER schema

3.4.2 WDBPALEO-SDB characterization

The WDBPALEO-SDB database system follow the same general rules for the WDBPALEO-IDB adaptation. In particular, some adjustment are required to store different type of paleoclimate information. In fact, the Nextdata research activities have been carried out to recover literature data (bibliography containing useful information for paleoclimatic studies) relating to the Holocene time interval, from which information about the last 2000 years were extracted. The research activities showed that data (biotic and abiotic proxies) from the marine sedimentary cores available for the last 2000 years are very few and characterized by a scattered geographical distribution. These data also associated with information from unpublished CNR-IAMC data were used to identify potential keysites in the Mediterranean Basin for the recovery of marine sediments containing sedimentary records of the last millennia. Additionally, the data available from some cores already collected by the CNR-IAMC in sites of interest (southern Tyrrhenian Sea, Gulf of Salerno) were reanalyzed and made available within the NextData project (Lirer, 2012).

The WDBPALEO-SDB system contains high resolution studies on the three cores (C90_1m-C90-C836) collected in 1998 and in 2006 on the continental shelf of the Gulf of Salerno (south-central Tyrrhenian Sea) at the depth of 103 meters. Marine sediment core data collected in SDB are the result of the following analysis and models output:

- Quantitative analysis of planktonic foraminifera (468 samples);
- Quantitative analysis of calcareous nannofossils (187 samples);

- Study on carbon and oxygen stable isotopes (468 samples) on the planktonic foraminifer species: Globigerinoides ruber;
- Tephrostratigraphic study (petrochemical analysis) on 8 levels of tephra;
- 8 AMS14C datings and 210Pb e 137Cs radionuclides dating of the first 40 cm of core.

Available of data and modeling output allow us to have the following paleoclimatic information through the SHARE Geonetwork web-system:

- Quantitative data on the distribution of planktonic foraminifera and calcareous nannofossils during the last 2000 years;
- δ18O e δ13C measurements data on the Globigerinoides ruber during the last 2000 years;
- Quantitative data on the distribution of the benthic foraminifera during the last 500 years;
- Petrochemical analysis of the 8 levels of tephra recognized in the C90_1m-C90-C836 composite core.

3.4.2.1 WDBPALEO-SBD main adaptations

Following the WDBPALEO-IDB adaptation directive SDB have been initialized:

- storing the data provider and adding geographical marine sediment core information (Fig. 3.12);
- adding new parameters representing the biotic and abiotic proxies from the marine sedimentary analysis;
- creating few tables, views and relation.

In spite of IDB, SDB have as main provider the CNR-IAMC, only.

The addition of new parameters has been developed grouping parameters depending on the analysis type:

- planktonic Foraminifera
- benthic Foraminifera

```
--SDBPALEO INIT--
SELECT wci.begin('sdbpaleo');

SELECT wci.addPerson(   'Filippo',
                        'Locci',
                        'Dr',
                        'Hi Filippo',
                        'FiL',NULL,
                        'male', NULL,
                        NULL,'2012-09-01','2999-12-31',NULL
                        );

INSERT INTO wdb_int.organizationtype VALUES ('non-profit association','A non-profit organization');

SELECT wci.addOrganization('Ev-K2-CNR – IAMC.CNR',
                                'Ev-K2-CNR – IAMC.CNR',
                                'non-profit association',
                                '1992-01-01','2999-12-31',
                                'NEXTDATA Project');

SELECT wci.addNameSpace(1111,
                                'Test SBD
namespace',
                                'namespace for SDB TESTING',
                                'production',
                                'Ev-K2-CNR – IAMC.CNR',
                                'FiL',
                                '2013-10-01'
                                );

SELECT wci.begin('sdbpaleo',1111,1111,1111);

SELECT wci.addDataProvider('IAMC.CNR',
                                'wci user',
                                'Point',
                                '1000 years',
'SeaCore Data from IAMC.CNR in the framework of NextData project'
                                );

--      PLACE NAME      LON             LAT
--      IAM00A076       14.7080         40.596
--      IAM00A077       14.7063         40.596
--      IAM00A078       14.7078         40.599

SELECT wci.addorupdateplacepoint('IAM00A076',
                                st_geomfromtext('POINT(14.7080 40.596)',4326),
                                '-infinity', 'infinity' );

SELECT wci.addorupdateplacepoint('IAM00A077',
                                st_geomfromtext('POINT(14.7063 40.596)',4326),
                                '-infinity', 'infinity' );

SELECT wci.addorupdateplacepoint('IAM00A078',
                                st_geomfromtext('POINT(14.7078 40.599)',4326),
                                '-infinity', 'infinity' );

select wci.copyparameternamespace(0);
```

FIGURE 3.12: SQL script for SDB initialization

- nannofossil
- radionuclides
- stable isotopes
- magnetic susceptibility
- secular variation of Earth's magnetic field

For each analysis a series of parameters are related. The following figure is an example of benthic Foraminifera parameters adding (Fig.3.13). Over than 60 parameters have been added for benthic Foraminifera alalysis.

As for the new tables a dating measurements table has been created to give information on core dating analysis (Fig. 3.14).

```
select wci.begin('sdbpaleo',1111,1111,1111);
select wci.addparameter('weight',NULL,NULL,NULL,NULL,NULL,NULL,'g');
select wci.addparameter('benthic forams total',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('benthic forams concentration',NULL,NULL,NULL,NULL,NULL,NULL,'1/g');
select wci.addparameter('adelosina sp',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('ammodiscus planorbis',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('ammonia beccarii',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('amphicoryna scalaris',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('asterigerinata mamilla',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('bigenerina nodosaria',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('bolivina alata',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('bolivina catanensis',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('bolivina spathulata',NULL,NULL,NULL,NULL,NULL,NULL,'none');
...
select wci.addparameter('uvigerina mediterranea',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('uvigerina peregrina',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('valvulineria bradyana',NULL,NULL,NULL,NULL,NULL,NULL,'none');
select wci.addparameter('core size',NULL,NULL,NULL,NULL,NULL,NULL,'m');
```

Figure 3.13: Benthic Foraminifera parameters adding script

placename	valid_data	valid_data_from	valid_data_to	corerangefrom	corerangeto	label	climaticevent	ecobiozone	note
iam00a076	268	188	348	55	66	tS1	Little	Ice	Age
iam00a077	268	188	348	39	48	tS1	Little	Ice	Age
iam00a077	1471	1421	1521	172	178	tS1-gamma	Roman	Period	1Fa
iam00a077	1471	1421	1521	216	295	tS2	Roman	Period	1Fa
iam00a077	3300	3200	3400	360	372	tS3	Middle	Bronze	Age
iam00a077	4500	4390	4280	412	420	tS4	Eneolithic	3F	Agnan
iam00a078	268	188	348	45	55	tS1	Little	Ice	Age
iam00a078	1471	1721	1521	170	180	tS1-gamma	Roman	Period	1Fa
iam00a078	1471	1721	1521	220	328	tS2	Roman	Period	1Fa
iam00a078	3300	3200	3400	390	391	tS3	Middle	Bronze	Age
iam00a078	3700	3590	3810	399	400	tS3-alfa	Middle	Bronze	Age
iam00a078	4500	4390	4610	431	441	tS4	Eneolithic	3F	Agnan
iam00a078	4500	4390	4610	539	540	tS4-alfa	Bölling-Allerod	7F	Neapoli

Figure 3.14: The new dating measurement table

```
select wci.begin('wdbpaleo',1111,1111,1111);
select wci.addcoredatingmeasurement(1,1111,
                                    'IAM00A076',
                                    1745,1665,1825,
                                    (55,66),'tS1',
                                    'Little Ice Age','1Fb',
                                    'Post AD 1631 activity');
--Function:
            wci.addcoredatingmeasurement(typeid_ integer,
                                         placename_ text,
                                         validtime_ real,
                                         datevalidfrom_ real,
                                         datevalidto_ real,
                                         corerange_ corelength, label_ text, climaticevent_ text,
                                         ecobiozone_ text, note text)
```

Figure 3.15: wci.addcoredatingdatingmeasurement example of use

To load data into the table wdb_int.coredatingmeasurement a new WCI function has been created, (wci.addcoredatingmeasurement) which check before data loading if the core point has been previously inserted (see Appendix B for details, Fig. B5). The following script show how to load dating measurements for the core named IAM00A076 (Fig. 3.15).

To load data into WDBPALEO-SDB system other few adjustment required. A new table named wdb_int.coreidentifiers has been created to store information about the core "section" and core "identifier". This table is related to the table wdb_int.floatvalueitem (see ER schema in Fig. 3.5). A new view has been created, wci_int.floatvalue_v useful in the retrieval data process. Finally, a new function to insert data in WDBPALEO-SDB has been created named wci.writepaleo substituting the original WCI function wci.write. See Appendix B for SQL code details Fig. B6 and Fig. B7.

Chapter 4

A comparison between MODIS Albedo Snow Product and in situ SHARE observations over the Himalaya Region: a case study

4.1 Introduction

In this Chapter a validation technique of the MODIS Snow Albedo product MOD10A1 using SHARE in situ observations over the Hymalaia region is illustrated. The technique involves entirely Open Source Resources, giving a clear and effective methodology to elaborate and operate on a large amount of data.

4.2 Relevant SHARE AWS data for remote sensing application

AWSs in monitoring high-altitude sites could be deployed over a wide range of surfaces, and have a variety of applications, including: climate variability assessment; in support of operational weather forecasting; model validation and in avalanche information support. Commonly, the measured variables include (see Appendix A Figure A.1 for sensors mounted in Ev-K2 CNR AWS):

- air and surface temperature

- wind speed and direction
- snow/ice accumulation and ablation
- humidity
- snow/ice conductive heat flux
- shortwave and terrestrial radiation fluxes
- atmospheric parameters (e.g. PM, HC, GHGs, BC)

Calibrations of satellite sensors, or comparison of satellite product retrievals with ground measurements are made often in large ice snow covering ((Stroeve and Nolin, 2002), (Stroeve et al., 2006)). Hence ground measurements, from AWS deployed on a glacier difficult to access, could be fundamental in comparison of remote sensing products to evaluate their accuracy and to improve retrieval algorithms itself ((Liang, 2000), (Pinty et al., 2011)). The AWS data measurements permit satellite comparison between albedo, temperature, thermal flux or net radiation.

4.3 Presentation of the case study

The starting assumption of this issue is the comparison between the shortwave snow albedo, calculated from shortwaves fluxes measured from the CNR1 ((Michel et al., 2007) radiometer mounted on Pyramid Laboratory- Observatory (Lobuche - Khumbu Valley) and the MODIS daily albedo snow product (MOD10A1). The case study involves the comparison betwen MODIS data and in situ measurements limited to few months (the pre-monson period from march to may) of the year 2006.

4.3.1 The MOD10A1 products overview

The **M**oderate **R**esolution **I**maging **S**pectroradiometer (MODIS) is a multispectral sensor mounted on the Earth Observing System (EOS)-AM1 (Terra) and EOS-PM1 (Aqua). The MODIS is a cross-track scanner, it can observe almost wholly the earth every two days, it is endowed by 36 spectral bands with 250m, 500m and 1km geometric instantaneous-fields-of-view (GIFOV) at nadir (Pape and Vohland, 2010). "The daily snow product is a tile of data gridded in the sinusoidal projection. Tiles are approximately 1200 x 1200 km (10degx10deg) in area. Snow data arrays are produced by selecting the most favorable observation (pixel) from the multiple observations mapped to a cell of the MOD10_L2G gridded product from the MOD10_L2 swath product. In addition to the snow data arrays mapped in from the MOD10_L2G, snow albedo is calculated. There are four Scientific Data Sets (SDSs or data fields) of snow data; snow cover map, fractional snow cover, snow albedo and QA in the data product file (see Table 4.1)" (Riggs et al., 2006). The MOD10A1 products are implemented in an HDF-EOS file. The Hierarchical Data Format (HDF) is the standard data format for all NASA Earth Observing System (EOS) data products https://nsidc.org/data/hdfeos/. The MOD10A1 uses an HDF internal compression to reduce the volume of the data files in the archive and the amount of network resources required to transport the data files. The snow product files or granules comprise:

- global attributes (metadata)
- SDSs i.e. data arrays with local attributes.

4.3.2 Total shortwave albedo

Total shortwave albedo is one key variable controlling the radiation energy budget of the land surface. Thus, the monitoring of its spatial and temporal variations is one important issue e.g. for the application of climate models (Pape and Vohland, 2010). Generally. albedo is defined as the ratio of reflected to incoming solar radiation at the surface. Hence, high reflecting means like snow and sea-ice covered surfaces have high albedo, unlike low reflecting means have low albedo values. In the process of comparison of two variables, in particular in situ observation and product estimates from remote sensing measurements, an accurate evaluation of each physical quantity and its meaning is fundamental. Because several definitions of albedo exist a clarification of terminology is required (Pape and Vohland, 2010). Becoming to distinguish between: Inherent albedo and apparent albedo.

- Inherent albedo (I_a) is a function of solar zenith angle, wavelength and surface properties. Then holding constant solar zenith angle and wavelength I_a depends only from the surface properties and thus is independent from the current atmospheric conditions.
- Apparent albedo (A_a) is defined as the ratio of upwelling and downwelling irradiance, and it is a function of the solar zenith angle and wavelength. The value of total downward radiance (direct and diffuse), accordingly the apparent albedo, is obviously a function of atmospheric conditions ((Liang, 2000), (Liang et al., 1999)).

Table 4.1: The MOD10A1 SDSs, (A. George et al, 2006)

Data Field Name	**Description**
Snow_Cover_Day_Tile	The snow cover map is the result of selecting the most favorable observation of all the swath level observations mapped into a grid cell for the day. Mapped is snow, snow-covered water bodies (typically lakes or rivers) land, water, cloud or other condition.
Fractional_Snow_Cover	The fractional snow cover map is the result of selecting the most favorable observation of all the swath level observations mapped into a grid cell for the day using the scoring algorithm. Fractional snow is reported in the 0 – 100% range, including inland water bodies. Pixels that are not snow are labeled as water, cloud or other condition.
Snow_Albedo_Daily_Tile	he snow albedo algorithm result is stored as a map of the snow albedo for the tile. The snow albedo map corresponds to snow mapped in the snow cover map in Snow_Cover_Day_Tile SDS. Snow albedo is reported in the 0 – 100 range and non-snow features are also mapped using different data values.
Quality Assessment	Spatial QA data corresponding to the snow cover observation selected for the daily snow cover map is also selected and mapped into the Snow_Spatial_QA SDS. Snow albedo specific QA is not reported in Collection 5 because ways of expressing the QA of the snow albedo result are being investigated. (Refer to the snow project website for validation information.) It is anticipated that future evaluation and validation of snow albedo will lead to the definition and setting of QA data. Fractional snow specific QA data is also not reported because evaluation and validation of the product has not been completed (refer to the snow project website for validation information).

A_a is equivalent to the inherent albedo only in absence of atmosphere. Taking into account these definitions albedometers or pyranometers measure A_a (Liang, 2000) like remote sensing sensors. Liang developed a method based on radiative transfer simulations to retrieve the broadband albedo from narrowband sensors. The Liang formula showed in the following equation (4.2) is used to compute MODIS total shortwave broadband albedo (α_{short}) (0.25-2.5μm) from its spectral albedos.

$$\alpha_{short} = 0.16\alpha_1 + 0.291\alpha_2 + 0.243\alpha_3 + 0.166\alpha_4 + 0.122\alpha_5 + 0.081\alpha_7 – 0.0015 \qquad (4.1)$$

Where α_i is the spectral albedo from spectral band i. In accordance with the Liang validation work (Liang et al., 2002) is straightforward to compare in situ albedo shortwave measurements and MODIS albedo products (MOD10A1 or MCD43A3). Accumulated short wave albedo (α_{Ashort}) is defined as the ratio of accumulated |SHW↑| and |SHW↓| over a time window of 24 h centered around the moment of observation (see 4.1).

$$\alpha_{Ashort} = \frac{\sum_{24h} |SHW \uparrow|}{\sum_{24h} |SHW \downarrow|} \qquad (4.2)$$

The 4.2 provides a useful measure of the effective albedo of the snow covered surface, and removes the effects from the varying solar zenith angle (Morin et al., 2012) Measurement of hourly SHW↑ and SHW↓ are provided by the CNR1-Radiometer mounted on Pyramid AWS.

4.3.3 Study area

The Pyramid Laboratory-Observatory is located (Nepal Climate Observatory - NCO-P, 5079 a.s.l.) on the Southern slope of the Himalayans(Fig 4.1). Notice that the NCO-P is the highest aerosol observatory managed within the Ev-K2-CNR Stations at High Altitude for Research on the Environment (SHARE) and the United Nations Environmental Program (UNEP) Atmospheric Brown Clouds (ABC) projects. The aerosol station was established in March 2006 for atmospheric research in the Khumbu Valley, Sagarmatha National Park, near the base of the Nepalese side of Mt. Everest (5079 m a.s.l.) (`http://evk2.isac.cnr.it/`). Because high altitude measurement sites are relatively clean and far from anthropogenic emission sources, they offer an opportunity to study the influence of anthropogenic pollution transported from remote areas (Yasunari et al., 2012). In fact a study using this data evaluate the decreasing of snow albedo caused by black carbon deposition using aerosol data of the year 2006 (Yasunari et al., 2012).

FIGURE 4.1: Location map of NCOP site. The square in orange denotes the NCOP site. http://reverb.echo.nasa.gov/

4.4 Material and methodology

The elaboration steps are three:

- Data download, WDB queries and albedo calculation
- Reprojection and Mosaic (using Python scripts and Modis Reprojection Tool (MRT))
- Data Processing and Mapping via Python Scripts.

Then MODIS data are downloaded and NCO-P radiative data are extracted from WDB system. To download MODIS data it is possible to use the REVERB system (http://reverb.echo.nasa.gov/reverb/) using the ftp script option.

After the data selection using the ftp command from the terminal, all selected files have been downloaded at once. A .txt file provided by REVERB system (Fig. 4.2) containing all information about granules selected is the input of the ftp command (see following example).

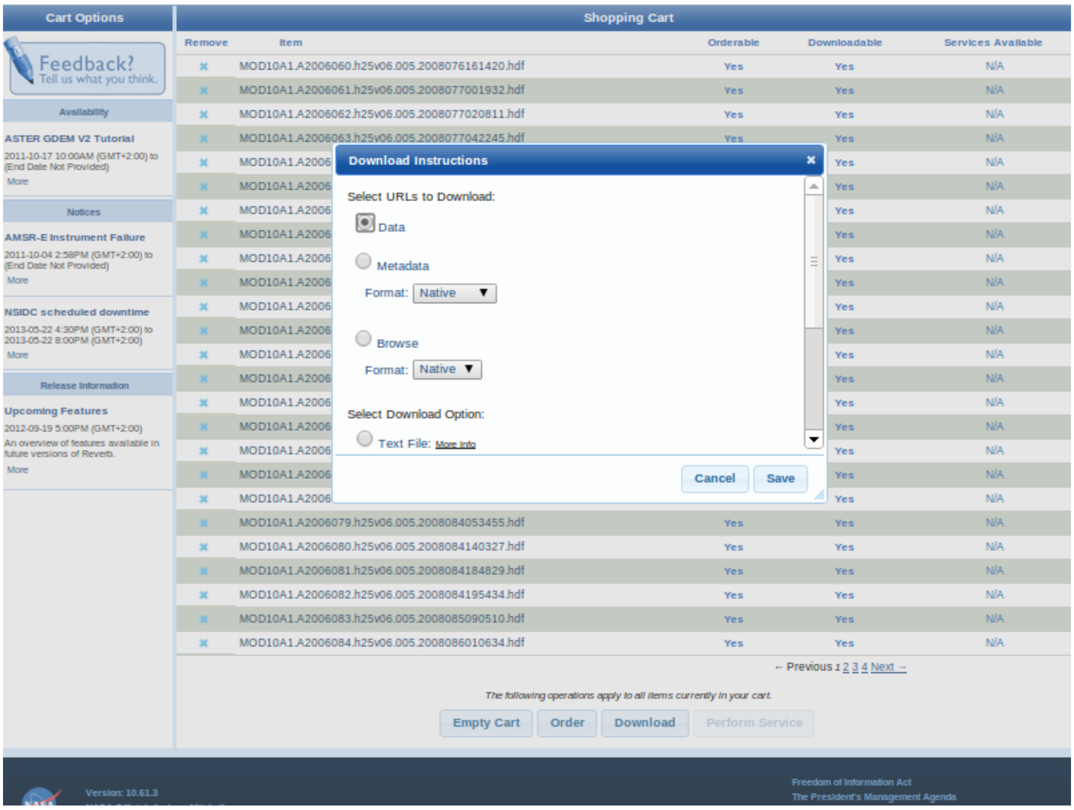


FIGURE 4.2: REVERB download screen-shot http://reverb.echo.nasa.gov/

```
ftp -p -n < data_url_script_2013-05-21_061751.txt
```

The α_{Ashort} is calculated starting from hourly shortwave radiation measured by NCO-P AWS (List. 4.1). Querying the WDB (List. 4.2) SHW↑ and SHW↓ have been retrieved and by the use of the Eq. 4.2, α_{Ashort} have been calculated (see Fig. 4.3) for the years from 2003 to 2010.

```
import string
import sys
import numpy as np
f=open('albedo_2002_2012_MATRIX.txt','rb')
g=open('albedo_2002_2012_CALC_TRUE.txt','w')
table_albedo = [row.strip().split('\t') for row in f]

dim_talbedo=len(table_albedo)

print table_albedo[0][1][0:10]
i=0
date_time=''
usw=0
date_time_usw=table_albedo[i][1]
```

```
dsw=0
date_time_dsw=table_albedo[i][3]

for i in range (0,dim_talbedo):
        if i!=dim_talbedo-1:
                if table_albedo[i][1][0:10]==table_albedo[i+1][1][0:10]:
                        usw=usw+float(table_albedo[i][0])
                        dsw=dsw+float(table_albedo[i][2])
                        date_time=table_albedo[i][3]
                else:
                        usw=usw+float(table_albedo[i][0])
                        dsw=dsw+float(table_albedo[i][2])
                        date_time=table_albedo[i][3]
                        albedo=float(usw)/float(dsw)
                        if albedo<=1:
                                g.write(date_time+'\t'+repr(albedo)+'\n')
                        usw=float(table_albedo[i][0])
                        date_time_usw=table_albedo[i][1]
                        dsw=float(table_albedo[i][2])
                        date_time_dsw=table_albedo[i][3]
        if i==dim_talbedo:
                        usw=usw+float(table_albedo[i][0])
                        dsw=dsw+float(table_albedo[i][2])
                        date_time=table_albedo[i][3]
                        albedo=float(usw)/float(dsw)
                        g.write(date_time+'\t'+repr(albedo)+'\n')
g.close()

g=open('albedo_2002_2012_CALC_TRUE.txt','rb')
date_table_albedo_calc=[]
data_table_albedo_calc=[]
table_albedo_calc = [row.strip().split('\t') for row in g]
print table_albedo_calc[0][1], len(table_albedo_calc)
for i in range (0,len(table_albedo_calc)):
        #a=np.array(table_albedo_calc[i][0],dtype='datetime64')
        #print a
        date_table_albedo_calc.append(table_albedo_calc[i][0][0:10])
print date_table_albedo_calc[0]
for i in range (0,len(table_albedo_calc)):
        #a=np.array(table_albedo_calc[i][0],dtype='datetime64')
        #print a
        data_table_albedo_calc.append(table_albedo_calc[i][1])
import time
from datetime import datetime
#t = datetime.now()
#t1 = t.timetuple()
pippo=datetime.strptime(date_table_albedo_calc[0], '%Y-%m-%d')
print pippo

import datetime as dt
dates=[datetime.strptime(ts, '%Y-%m-%d') for ts in date_table_albedo_calc]
#dates=[dt.datetime.fromtimestamp(ts) for ts in date_table_albedo_calc]
from matplotlib import pyplot
import matplotlib.pyplot as plt
import matplotlib.dates as md
```

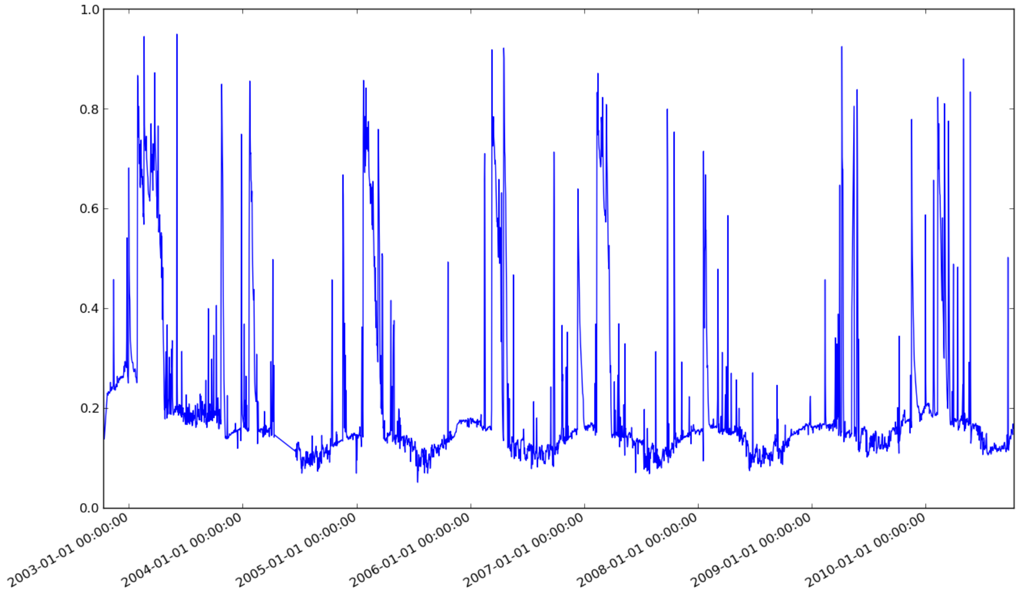


FIGURE 4.3: Accumulated SW albedo calculated for the years 2002-2010

```
ax=plt.gca()
xfmt = md.DateFormatter('%Y-%m-%d %H:%M:%S')
ax.xaxis.set_major_formatter(xfmt)
plt.plot(dates, data_table_albedo_calc)
plt.gcf().autofmt_xdate()
plt.show()
```

Listing 4.1: Python script to calculate α_{Ashort} and to graph Fig. 4.3

```
SELECT wci.begin('wdb',1000,1000,1000);
SELECT * FROM wci.read ( ARRAY['ev-k2-cnr committee'],
                        'pyramid laboratory observatory share',
                        'inside 2002-01-01 TO 2012-12-31',
                        NULL,
                        ARRAY[  'hourly min USW radiation',
                                'hourly average USW radiation',
                                        'hourly max USW radiation',
                                        'hourly average USW radiation',
                                        'valid hourly USW radiation',
                                        'USW radiation'],
                                         NULL,NULL,NULL::wci.returnfloat);
```

Listing 4.2: WDB query for SHW↑ and SHW↓ retrieving

The second step consists of reading each .hdf file and create for each of them a .prm file that is the input for the MRT **resample** function. The following Python script (List. 4.3) produce the .prm file. The .prm file passes the information to the **resample** function to extract from the .hdf file a tile centerend around NCO-P:

```
prm.write('SPATIAL_SUBSET_TYPE = INPUT_LAT_LONG\n')
prm.write('SPATIAL_SUBSET_UL_CORNER = ( 30 85 )\n')
prm.write('SPATIAL_SUBSET_LR_CORNER = ( 27 89 )\n')
```

with the specified SPECTRAL SUBSET, flagging with 1 to extract the SUBSET (Snow Cover Daily Tile Field, Snow Albedo Daily Tile Field, Snow Spatial QA Field, Fractional Snow Cover Field) or 0 to skip extraction .

```
prm.write('SPECTRAL_SUBSET = ( 1 1 1 1 )\n')
```

In fact, MOD10A1 product consists of 2400 x 2400 cells of tiled data in a sinusoidal projection. Each data granule contains the following HDF-EOS local attribute fields, which are stored with their associated Scientific Data Set (SDS):

- Snow Cover Daily Tile Field
- Snow Albedo Daily Tile Field
- Snow Spatial QA Field
- Fractional Snow Cover Field

```
import sys
import os
import string

os.system("ls *.hdf > hdf.txt")
hdf_f=open('hdf.txt','r')
file_n=hdf_f.readline()
i=0
while file_n:
        f=file_n.replace('\n','')
        f=f.replace(' ','')
        print i,f
        prm=open(f+'.prm','w')
        input_row='INPUT_FILENAME = '+f
        output_row='OUTPUT_FILENAME = '+ f[0:42]+'tif'
        prm.write(input_row+'\n')
        prm.write('SPECTRAL_SUBSET = ( 1 1 1 1 )\n')
        prm.write('SPATIAL_SUBSET_TYPE = INPUT_LAT_LONG\n')
        prm.write('SPATIAL_SUBSET_UL_CORNER = ( 30 85 )\n')
        prm.write('SPATIAL_SUBSET_LR_CORNER = ( 27 89 )\n')
        prm.write(output_row+'\n')
```

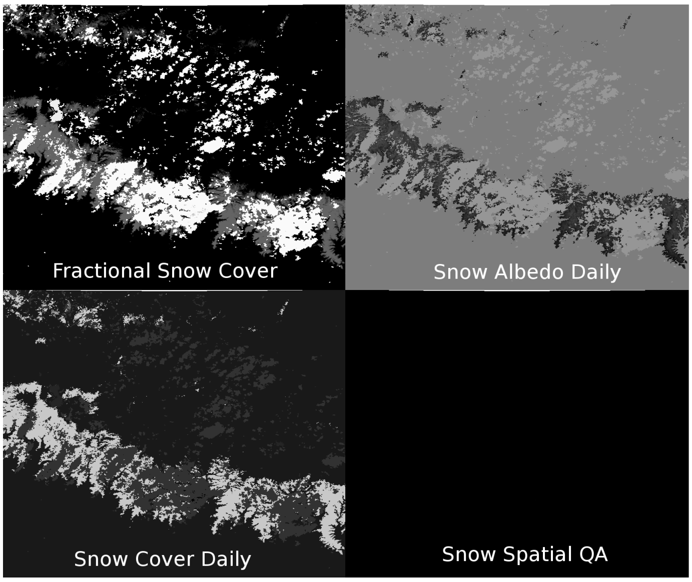


FIGURE 4.4: GEOTIFF images produced by **resample** function

```
        prm.write('RESAMPLING_TYPE = NEAREST_NEIGHBOR\n')
        prm.write('OUTPUT_PROJECTION_TYPE = UTM\n')
        prm.write('OUTPUT_PROJECTION_PARAMETERS = (\n')
        prm.write('0.0  0.0  0.0\n')
        prm.write('0.0  0.0  0.0\n')
        prm.write('0.0  0.0  0.0\n')
        prm.write('0.0  0.0  0.0\n')
        prm.write('0.0  0.0  0.0 )\n')
        prm.write('DATUM = WGS84\n')
        prm.write('UTM_ZONE = 45\n')
        prm.write('OUTPUT_PIXEL_SIZE = 463.3\n')
        prm.close()
        file_n=hdf_f.readline()
        i=i+1
hdf_f.close()
```

Listing 4.3: Python script to produce .prm files

The following batch script (List. 4.4) reads all .hdf files and the related .prm files previously produced and generate a geotiff image (Fig. 4.4) of the selected SUBSET by the use of the MRT **resample** function:

```
#!/bin/bash
        for i in *.hdf
        do
        echo $i
        resample -p $i.prm
        done
```

Listing 4.4: Bash script using the MRT resample function

Table 4.2: Local Attributes for Snow Cover and Snow Albedo

Value and Attribite	**Product**
0-100=snow albedo in percent	Snow_Albedo_Daily_Tile
200 = snow snow-covered land	Snow_Cover_Daily_Tile

The third step comprises the pixels extraction from the geotiff images previously produced relating to each pixel the local attribute. Two products are considered Snow Cover and Snow Albedo Tile: the first product is used to discriminate if each extracted pixel is an effective snow pixel. In brief, for each extracted pixel a .txt file is created with its attribute, relating each snow albedo pixel with the corresponding snow cover pixel. The following Python (get_pixel_all.py, snow_albedo_get.py and snow_cover_get.py) scripts illustrate how to extract the pixels centered around NCO-P and build the .txt files with Snow Cover and Albedo data and related attributes.

The **get_pixel_all.py** script (List. 4.5) gets the pixels from geotiff images in the neighborhood of an input (lat,lon) coordinates. The neighborhood is defined by an entered radius from the center (input coordinates). The pixels are selected if their coordinates are included in the entered radius:

```
if abs(Decimal(la)-Decimal(latP[0]))<=0.005 and
abs(Decimal(lo)-Decimal(lonP[0])) <=0.005:
```

The entered radius (0.005) select four pixel centered around the input coordinates (latP,lonP). Increasing the size of the radius the number of the selected pixels increase.

```
#GET PIXEL
from mpl_toolkits.basemap import Basemap,cm
import gdal
from osgeo import osr, gdal
from gdalconst import *
import numpy as np
import matplotlib.pyplot as plt
import pyproj
import os
import string

# read the data

#READ FROM STDIN THE GEOTIFF FILE
os.system("ls *.tif > tif.txt")
f=open('tif.txt','r')
file_im=f.readline()
while file_im:
        #file_im=raw_input('Inserisci file tif da elaborare: ')
```

```
file_im=file_im.replace('\n','')
data_matrix=open(file_im+'.txt','w')
print repr(file_im)
print data_matrix
#raw_input('inserisci file tif ...: ')
#dataset is the GEOTIFF object, assigned using gdal library, using Open
dataset = gdal.Open(file_im,GA_ReadOnly)

#band1 is an array composeb by the rastedata from GEOTIFF
band1 = dataset.GetRasterBand(1).ReadAsArray()

#Get coordinate from dataset
pr=dataset.GetProjectionRef()
GT = dataset.GetGeoTransform()

#Pyproj library is used to trasform XY coordinate in lat lon coordinate
#using a specific projection (here Mercator WGS84)
pp = pyproj.Proj(proj="utm",zone=45,ellps='WGS84')

#Lower Left  ( 1466939.882, 2914096.065) ( 84d38' 8.12"E, 26d 1'21.81"N) zona 43
#Upper Right ( 1706465.982, 3380175.865) ( 87d27'15.27"E, 29d57'25.81"N)
#Lower Left  (  301545.766, 2987819.887) ( 85d 0' 0.11"E, 26d59'53.89"N)zona 45
#Upper Right (  698593.866, 3320469.287) ( 89d 3'31.83"E, 29d59'56.73"N)



lllon,lllat=pp(301545.766, 2987819.887, inverse=True)
urlon,urlat=pp(698593.866, 3320469.28, inverse=True)

print lllon
print lllat
print urlon
print urlat

lat = np.linspace(urlat, lllat, band1.shape[0])
lon = np.linspace(urlon, lllon, band1.shape[1])

Lon, Lat = np.meshgrid(lon, lat)

#Pyramid coord
latP=[27.9588888888889]
lonP=[86.8127777777778]
coordPy=[latP,lonP]
print GT
print band1.shape
print pr



import commands
from decimal import *
getcontext().prec = 2

shellout = commands.getoutput('gdalinfo '+file_im)
a=shellout.split('\n')
```

```
print a[29],'\n', a[30]
print a[31],'\n', a[32]
print a[33],'\n', a[34]

ll=a[29].split(' ')
print ll
count=0
print len(Lon),' ',len(Lat)

for i in range(0,band1.shape[0]-1):
        for j in range(0,band1.shape[1]-1):
                a=[Lat[i,j],Lon[i,j]]
                la=Lat[i,j]
                lo=Lon[i,j]
                #print Decimal(la)-Decimal(latP[0])
                #print Decimal(lo)-Decimal(lonP[0])
                #v=raw_input()
                if abs(Decimal(la)-Decimal(latP[0]))
                                <=0.005 and abs(Decimal(lo)-Decimal(lonP[0])) <=0.005:
                        count=count+1
                        data_matrix.write(repr(i)+' '+repr(j)+' '+repr(count)
                                +' '+repr(a)+' '+repr(band1[i,j])+'\n')
                        print i,' ',j,' ',count, ' ',repr(a),' ',band1[i,j]
                #raw_input()
                j=j+1
        i=i+1
file_im=f.readline()
```

Listing 4.5: It Gets the pixels from geotiff images

The scripts **snow_albedo_get.py** (List. 4.7) and **snow_cover_get.py** (List. 4.6) for each pixel related their "local attributes" (Fig. 4.5).
(http://nsidc.org/data/docs/daac/mod10_modis_snow/version_5mod10a1_local_attributes.html#snowcoverdailytilefield).

```
import os
import string
#####################################################################
#Custom Local Attributes for the Snow_Cover_Daily_Tile Field
#Sample Value
#missing_value  Coded integer used to indicate missing data.    0
#Key    Key to the meaining of the coded integers within the SDS.
#Value                          #Description
#0 = missing data               data missing
#1 = no decision                no decision
#11 = night                     darkness, terminator, or polar night
#25 = no snow                   snow-free land
#37 = lake                      lake or inland water
#39 = ocean                     open water
#50 = cloud                     cloud obscured
#100 = lake ice                 snow-covered lake ice
#200 = snow                     snow-covered land
```

```
#254 = detector saturated       detector saturated
#255 = fill                     fill
####################################################################
codifica={'0':'missing data',
'1':'no decision',
'11':'night',
'25':'no snow-free land',
'37':'lake',
'39':'ocean',
'50':'cloud',
'100':'snow-covered lake ice',
'200' :'snow-covered land',
'254':'detector saturated',
'255':'fill'
}
#READ FROM STDIN THE GEOTIFF FILE
os.system("ls M*.Snow_Cover_Daily_Tile.txt > Snow_Cover_Daily_Tile_list.txt")
f=open('Snow_Cover_Daily_Tile_list.txt','r')
g=open('Snow_Cover_Daily_Tile.txt','w')
g.write('DATA\t\tPixel\t\tValore\t\tDescrizione\n')
file_SC=f.readline()
while file_SC:
        file_SC=file_SC.replace('\n','')
        data=file_SC[8:16]
        print file_SC
        print data
        #raw_input()
        h=open(file_SC,'r')
        snow_c=h.readline()
        riga=snow_c.split(' ')
        while snow_c:
                riga=snow_c.split()
                pixel=riga[2]
                valore=riga[5]
                print codifica[valore]
                g.write(data+'\t\t'+pixel+'\t\t'+valore+'\t\t'+codifica[valore]+'\n')
                print snow_c
                print riga
                snow_c=h.readline()

        file_SC=f.readline()
```

Listing 4.6: The script to get snow cover attribute

```
####################################################################
#Custom Local Attributes for the Snow_Albedo_Daily_Tile Field

#Value                          Description
#0-100 = snow                   albedo in percent
#101 = no decision              no decision
#111 = night                    darkness, terminator, or polar night
#125 = land                     snow-free land
#137 = inland water             lake or inland water
#139 = ocean                    open water
#150 = cloud                    cloud obscured
```

```
#250 = missing                      data missing
#251 = self_shadowing               self shadowing
#252 = landmask mismatch            landmask mismatch
#253 = BRDF_failure                 Bidirectional Reflectance Distribution Function failure
#254 = non-production_mask          non-production mask
######################################################################
codifica={repr(i):'albedo' for i in range(0,101)}
codifica.update({'0':'missing data',
'101':'no decision',
'111':'night',
'125':'no snow-free land',
'137':'lake or inland water',
'139':'ocean',
'150':'cloud',
'100':'snow-covered lake ice',
'250' :'missing',
'251' :'self shadowing',
'252':'landmask mismatch',
'253':'BRDF failure',
'254':'non production mask',
'255':'fill',
})
#READ FROM STDIN THE GEOTIFF FILE
os.system("ls M*.Snow_Albedo_Daily_Tile.txt > Snow_Albedo_Daily_Tile_list.txt")
f=open('Snow_Albedo_Daily_Tile_list.txt','r')
g=open('Snow_Albedo_Daily_Tile.txt','w')
g.write('DATA\t\tPixel\t\tValore\t\tDescrizione\n')
file_SC=f.readline()
print codifica;
raw_input()
while file_SC:
        file_SC=file_SC.replace('\n','')
        data=file_SC[8:16]
        print file_SC
        print data
        #raw_input()
        h=open(file_SC,'r')
        snow_c=h.readline()
        riga=snow_c.split(' ')
        while snow_c:
                riga=snow_c.split()
                pixel=riga[2]
                valore=riga[5]
                print codifica[valore]
                g.write(data+'\t\t'+pixel+'\t\t'+valore+'\t\t'+codifica[valore]+'\n')
                print snow_c
                print riga
                snow_c=h.readline()

        file_SC=f.readline()
```

Listing 4.7: The script to get the albedo values

DATA	Pixel	Valore	Descrizione
2006-03-01-05:20:00.000000 to 05:30:00.000000	1	125	no snow-free land
2006-03-01-05:20:00.000000 to 05:30:00.000000	1	125	no snow-free land
2006-03-01-05:20:00.000000 to 05:30:00.000000	1	125	no snow-free land
2006-03-01-05:20:00.000000 to 05:30:00.000000	1	125	no snow-free land
2006-03-01-05:20:00.000000 to 05:30:00.000000	1	125	no snow-free land
2006-03-02-04:25:00.000000 to 06:10:00.000000	2	125	no snow-free land
2006-03-02-04:25:00.000000 to 06:10:00.000000	2	125	no snow-free land
2006-03-02-04:25:00.000000 to 06:10:00.000000	2	125	no snow-free land
2006-03-02-04:25:00.000000 to 06:10:00.000000	2	125	no snow-free land
2006-03-02-04:25:00.000000 to 06:10:00.000000	2	125	no snow-free land
2006-03-03-05:05:00.000000 to 05:15:00.000000	3	125	no snow-free land
2006-03-03-05:05:00.000000 to 05:15:00.000000	3	125	no snow-free land
2006-03-03-05:05:00.000000 to 05:15:00.000000	3	125	no snow-free land
2006-03-03-05:05:00.000000 to 05:15:00.000000	3	125	no snow-free land
2006-03-03-05:05:00.000000 to 05:15:00.000000	3	125	no snow-free land
2006-03-04-04:10:00.000000 to 06:00:00.000000	4	125	no snow-free land
2006-03-04-04:10:00.000000 to 06:00:00.000000	4	125	no snow-free land
2006-03-04-04:10:00.000000 to 06:00:00.000000	4	125	no snow-free land

FIGURE 4.5: The extracted Snow Albedo data with their attributes

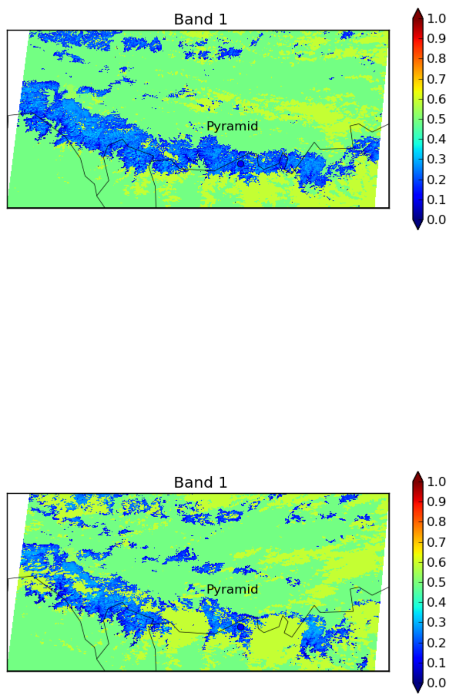


FIGURE 4.6: Snow albedo map elaboration from geotiff imaged

4.5 Results

The following images represent two albedo maps, generated using the geotiff images, illustrating a snow free day and a snow covered day (Fig. 4.6). The script List. 4.8 plots the albedo maps.

```
from mpl_toolkits.basemap import Basemap,cm
import gdal
from osgeo import osr, gdal
```

```
from gdalconst import *
import numpy as np
import matplotlib.pyplot as plt
import pyproj

# read the data

#READ FROM STDIN THE GEOTIFF FILE
file_im=raw_input('inserisci file tif ...: ')

#dataset is the GEOTIFF object, assigned using gdal library, using Open
dataset = gdal.Open(file_im,GA_ReadOnly)

#band1 is an array composeb by the rastedata from GEOTIFF
band1 = dataset.GetRasterBand(1).ReadAsArray()

#Get coordinate from dataset
pr=dataset.GetProjectionRef()
GT = dataset.GetGeoTransform()

#Pyproj library is used to trasform XY coordinate in lat lon coordinate
#using a specific projection (here Mercator WGS84)
pp = pyproj.Proj('+proj=merc +lon_0=0 +k_0=1.0 +x_0=0 +y_0=0')


#get the coordinates in lat long

#latlong = transform.TransformPoint(minx,miny)

#Lower Left  ( 9016878.754, 2376446.579) ( 81d 0' 0.00"E, 20d59'56.17"N)
#Upper Right ( 9239726.054, 3228918.579) ( 83d 0' 6.74"E, 28d 0' 0.00"N)
#Lower Left  ( 9016878.754, 2376291.985) ( 81d 0' 0.00"E, 20d59'51.47"N)
#Upper Right ( 9477398.954, 3482189.085)
#Lower Left  ( 9016878.754, 3228763.985) ( 81d 0' 0.00"E, 27d59'55.56"N)
#Upper Right ( 9484811.754, 3482189.085) ( 85d12'12.65"E, 30d 0' 0.00"N)
#Lower Left  ( 9016878.754, 3103672.985) ( 81d 0' 0.00"E, 26d59'48.25"N)
#Upper Right ( 9462573.354, 3482189.085) ( 85d 0'13.47"E, 30d 0' 0.00"N)
#Lower Left  ( 9462156.717, 3103672.985) ( 85d 0' 0.00"E, 26d59'48.25"N)
#Upper Right ( 9996804.917, 3482189.085) ( 89d48'10.18"E, 30d 0' 0.00"N)



lllon,lllat=pp(9462156.717, 3103672.985, inverse=True)
urlon,urlat=pp(9996804.917, 3482189.085, inverse=True)

print lllon
print lllat
print urlon
print urlat

#lat and lon are two spaced array giving dimension of GEOTIFF image
#band1.shape[0],band1.shape[1] are the dimension. linspace method is
#used to fom these spaced arrays.
lat = np.linspace(urlat, lllat, band1.shape[0])
lon = np.linspace(urlon, lllon, band1.shape[1])
```

```
#Lon and Lat provide the grid where pixels are plotted, formed
#by previous lat and lon arrays.
Lon, Lat = np.meshgrid(lon, lat)

# make the base map
m = Basemap(projection="merc", resolution="l",
                        llcrnrlon=urlon, urcrnrlon=lllon,
                        llcrnrlat=lllat, urcrnrlat=urlat)
m.drawcoastlines()
m.drawstates()
m.drawcountries()
# draw parallels and meridians
m.drawparallels(np.arange(lllon,urlon,1),labels=[1,1,0,0])
m.drawmeridians(np.arange(lllat,urlat,1),labels=[0,0,0,1])
.
# compute native map projection coordinates of lat/lon grid.

x, y = m(Lon,Lat)

# plot data over the map
cs = m.pcolor(x,y,band1,cmap=plt.cm.jet)
# plot colorbar over the map
cb = plt.colorbar(cs, shrink=0.6, extend='both')
plt.title(" Band 1 ")

# plot the names of the citiy.
latP=[27.96]
lonP=[86.81]
cities=['Pyramid']
xp,yp=m(lonP,latP)
m.plot(xp,yp, 'o', markersize=6)
for name,xpt,ypt in zip(cities,xp,yp):
    plt.text(xpt+50000,ypt+50000,name)

#save the map
plt.savefig('/home/filippo/Scrivania/basemap.png')

#plt.show()
```

Listing 4.8: The script used to plot albedo Maps

The comparison between albedo data from in situ measurements and MODIS data uses the results from previous sections. The days for comparison have been chosen getting only the pixels with the attribute snow cover. The number of the days are reduced drastically for the pre-monson period over the months from march to may of 2006. Many days in this period are not cloudy free over NCO-P, and then they were discarded. Available data for the comparison are 4 days of April (from 14^{th} to 17^{th}) 2006 and 8 days of May (from 8^{th} to 15^{th}). In the next figure (4.7) the scatter plot and a regression line between calculated albedo data MODIS albedo. In brief, in the following table (4.3) the related statistics.

Table 4.3: Statistics

Descriptor	**Value**
Mean Short Wave Albedo NCO-P	0.1365088294
Sample Standard Ddeviation of SWA-NCOP	0.0223376356
Mean of MOD10A1 Albedo	0.2841666667
Sample Standard Ddeviation of MOD-A	0.1036127698
Correlation of SWA-NCOP and MOD-A	0.5019150339
Slope of the regression line is	2.44543660875
Intercept is	-0.0365499604693

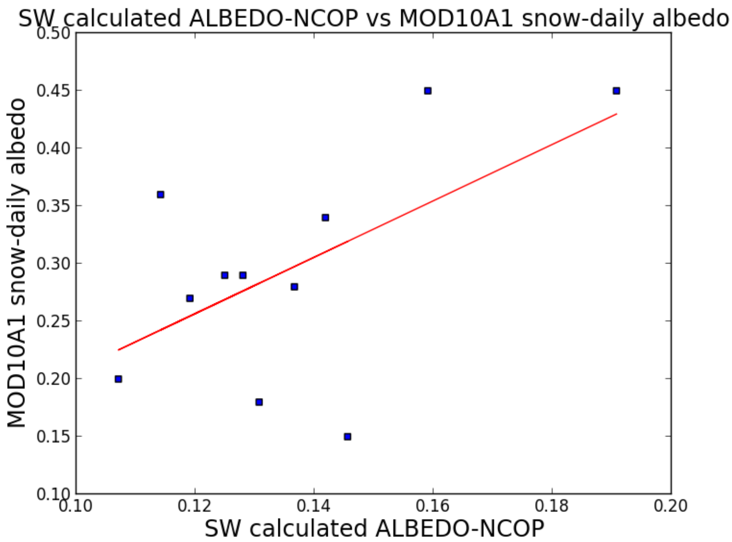


Figure 4.7: Snow albedo vs MOD10A1 albedo scatter plot

A moderate correlation exists (0.50) between the calculated albedo and the MOD10A1 albedo product. From data analysis MOD10A1 snow albedo daily is overestimated of about 14% compared to calculated short wave albedo measurements. In the last chapter the results will be discussed in depth.

Chapter 5

Discussion and conclusions

5.1 Introduction

This section wishes to show the main characteristics of the EDS while discussing its strength, critical aspects and future development. Then it discusses the role of data extracted directly from the EDS for scientific purposes, in particular for the comparison with remote sensing data using Open Sources resources. Two different discussions will be treated distinguishing between the EDS and the use of the EDS data in a particular case study.

5.2 The EDS: a critical overview and future perspectives

This work has focused its efforts on the need for cataloging the metadata related to:

- high elevation environmental and territorial data
- meteoclimatic and atmospheric composition data
- past climate information from ice and sediment cores
- biodiversity and ecosystem data
- measurements of the hydrological cycle
- marine reanalyses and climate projections at global and regional scale

in a comprehensive Environmental Data System. The SHARE Geonetwork (SGN) web platform have been developed to get access to the complete set of data and metadata collected in the frameworks of the SHARE (http://www.evk2cnr.org/cms/en/

share/project/intro) and NextData (http://www.nextdataproject.it/) project. The SGN web-platform will provide basically three types of services:

- structured metadata archive, data and results from remote mountain and marine areas researches and projects
- access to: high-altitude SHARE stations for the creation of a network of existing stations; marine sediment core and non polar ice core data archive developed for the NextData project.
- dedicated WEBGIS for geo-referenced data collected during the research.

The huge amount of information available in this web-platform is organized by themes and macro categories of reference, corresponding to the scientific disciplines of interest (see Fig. 5.1 (A) (B)):

- Stations at High Altitude
- Atmosphere and Climate
- Ecosystems and Biodiversity
- Geology and Geophysics
- Glaciers
- Ground data
- Health
- Paleoclimate
- Satellite Images

Moreover, the user can access directly to the SHARE and NextData resources (Fig. 5.1 (C)) (data and metadata) using dedicated categories:

A system of hierarchical privileges, roles, and user groups to manage users and permissions to access, modification and data downloading have been designed. For the access to public information there are no restrictions, while in order to have access to specific information or functionality, an account that will be provided by the system administrator will be required. Also, it will be possible to read the information about a resource, download or browse interactively data for that resource depending on the role of an authenticated user and privileges set for the metadata record. Authenticated

(A) Share Categories

(B) NextData Categories

(C) SHARE and NextData metadata access

FIGURE 5.1: The SGN resources access

users (and depending on privileges) can create, import and edit metadata records. They can also upload and configure data links for interactive map. Each authenticated user is assigned to a particular working group and it is able to view the data within that group (Fig. 5.2). The core of the information system is installed in a server located at the Ev-K2 CNR Committee Centre in Bergamo (main node), but there is the possibility to install relocated subsystems based on the same technology, named focal point of SHARE, which will contain metadata and data connected to the main node. Each focal point have its own region of interest and the system will be able to make a search on all nodes simultaneously. This distributed system enhance the efficiency in term of speed of access to resources. In Geonetwork open source this feature is called harvesting: with this function it is possible to find specific information residing on different nodes installed worldwide and periodically copy and store this information locally.

In this way an user from a single point of access may also obtain information from distributed catalogs. Through the technology of harvesting could be scheduled periodically the reading of information from databases located in different parts of the world and as a result selection of data obtained from a survey distributed in a single Web page return. In order to extend the network with the partnership of other data provider the EDS is implemented taking into account the input of new metadata sets and new data

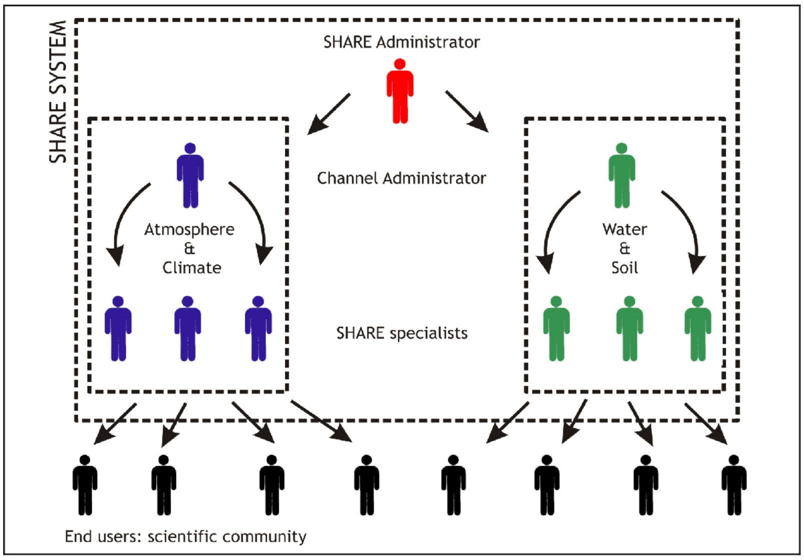


FIGURE 5.2: SHARE Geonetwork user hierachy

formats. In fact, for each new remote mountain weather station, ice/sediment core it is possible to translate the native data format in accord to the metadata set required in WDB/WDBPALEO. It is in planning to develop a graphical query queries in netCDF format.

5.2.1 First conclusions

SHARE, NextData and the SHARE Geonetwork project in particular, were born from the necessity to improve the knowledge of mountains and marine ecosystems and sharing data and metadata with the scientific community. This information are collected by direct observations from climatic and meteorological high mountain stations, and from ice/sediment core analysis. The salient features of this project is access to and collection of research data in sensitive and remote areas, carried out in collaboration with local and governmental institution for the aim of preserving these fragile ecosystems (Salerno et al., 2010). The SHARE Geonetwork project, in particular, provides: a further impetus to high mountains research, complementing other projects such as GLORIA (Global Observation Research Initiative in Alpine Environments) (Grabherr et al., 2000) and the recent creation of a database for storing, processing and sharing glaciological data in Italy (Nigrelli and Marino, 2012); a methodology to implement a climatological/geographical information system using open source tools. At the same time, the NextData project, promoted by the Italian Ministry for Education, University and Research, coordinated by the CNR Department of Earth and Environment, which the aim is the measurement, interpretation and provision of environmental and climatic data of high altitude areas, it is designed to obtain information on natural climate variability over the last thousand years, to quantify changes underway and to develop future scenarios for mountain regions. Finally, the next planned activity within the Geonetwork SHARE will be to integrate the portal I-AMICA, a project coordinated by CNR and developed under the Convergence Objective, cofunded by the European Regional Development Fund (ERDF) and the Italian Ministry of Education, University and Research, and the Italian Ministry of Economy and Finance. Through the implementation of the Geonetwork infrastructure, I-AMICA aims to encourage the development of specific Italian regions of the Mediterranean, also through a platform that facilitates the convergence of information gathered, necessary to achieve strategic objectives and harmonize policies for access and use of data, useful to strategies for technological adaptation and innovation.

5.3 EDS's data use: discussion about results and methodology

The case study presented in Chapter 4 is helpful to illustrate the importance of AWS data, especially in remote areas (like glaciers and high mountains), where remote sensing products and algorithms are difficult to validate. Only shortwave and terrestrial radiation fluxes are being considered in this case, but other physical variable are recorded during years by SHARE AWSs, like air and surface temperature or humidity, useful for other comparison or elaborations. It is important to note that the meteorological data from AWS were validated in two different way: from 2002 to 2009 the datum followed the guidelines defined within the CEOP criteria (Coordinated Energy and Water Observation Project) (CEO), whereas, from 2010 to 2011, the validation process took place under WMO recommendations on quality control and data validations, as WMO reference document (Zahumensky, 2004). The obtained results, in particular the moderate correlation value (about 0.5) motivate us to further data analysis. In brief, it will be necessary:

- to extend the temporal analysis up to five or ten years, enhancing the statistical significance
- to includes physical analysis in the results interpretation (e.g. including the solar zenith angle influence; considering the footprint of the device is about 300 m2 (the instrument is about 1 m above the surface) (Kipp-Zonen, 2009) for a geostatistical analysis; evaluating the slope influence on the albedo estimate and calculation)
- to extend the comparison with other physical parameters (e.g. soil temperature, humidity)

Date	ANCOP	AMODIS	Amodis-Ancop
14/04/06	0,190720669	0,45	0,2592793307
15/04/06	0,159028914	0,45	0,2909710857
16/04/06	0,141823762	0,34	0,1981762385
17/04/06	0,136613452	0,28	0,143386548
08/05/06	0,124906647	0,29	0,1650933532
09/05/06	0,114127702	0,36	0,2458722976
10/05/06	0,119052699	0,27	0,150947301
11/05/06	0,127956081	0,29	0,1620439189
12/05/06	0,130685204	0,18	0,0493147964
13/05/06	0,107038123	0,2	0,0929618768
14/05/06	0,145550118	0,15	0,0044498822
15/05/06	0,140602582	0,15	0,0093974175

Corr.	Corr. Apr	Corr. May1	Corr. May2
0,501915034	0,832471681	-0,958472051	-0,6416456785
	Mean	Mean	Mean
NCOP	0,157046699	0,130969007	0,1215107823
MOD	0,28	0,17	0,3025
	St. Dev.	St. Dev.	St. Dev.
NCOP	0,024407109	0,017109279	0,0061542889
MOD	0,084459063	0,024494897	0,0394757309

FIGURE 5.3: Statistical albedo correlation analysis

In order to improve the results of this study the data were subdivided into three groups on the base of the acquisition date. The correlation between these new groups of data appear increased (see Fig. 5.3).

This results seems to confirm the influence of solar zenith angle in the MOIDIS snow albedo retrieval (Liu et al., 2009). It will be necessary further statistical evidences to confirm this correlation.

Appendix A

SHARE Network Measuring Sensors

In the next figures are summarized the sensors mounted in each station and laboratories, installation date and time interval of records parameters. In particular Figure A.1 lists the meteorological parameters and Figure A.2 the atmospheric parameters. Each station starts keeping a record of observations from the "Installment Date" until the present. The raw data keep track of whole data since the "Installation data".

STATION NAME	SENSORS														
	Thermohygrometer	Soil moisture sensor	Barometer	Combined wind speed-direction sensor	CNR1 net radiometer	Electric rain gauge	Geothermometer	Soil heat flux sensor	Pyranometer	UVA radiometer	Snow depth sensor	Anemometer	Geothermometer	Heat Flux sensor	Installation Date
Forni	x		x	x	x						x				26/09/05
Dosdé	x		x		x						x				14/08/07
Mt. Bianco- OSRAM"	x		x		x						x				17/12/07
Namche	x		x	x		x			x						27/10/01
Lukla	x	x	x	x	x	x	x	x	x				x		01/09/01
Pheriche	x		x	x		x			x						15/10/01
Pyramid Laboratory- Observatory	x		x	x	x	x			x		x		x	x	2000/2004
Kala Patthar	x		x	x		x			x	x					15/05/08
Changri Nup Glacier	x			x	x										15/02/10
Mt. Everest – South Col	x		x						x	x		x			05/05/08
Askole (Baltistan, Pakistan)	x		x	x		x			x						07/10/05
Urdukas	x		x	x		x			x						17/06/04
Rwenzori - Elena Glacier	x		x	x		x			x						18/07/06

Figure A.1: List of meteorological parameters for each station

Sensors	Mt. Cimone	Climate Ob. Pyramid
Condensation Particle Counter	x	-
Differential Mobility Particle Sizer (ASD (10 nm – 500nm)	x	-
Ultrasonic anemometer	x	-
Optical Particle Counter ASD(300 nm – 20 µm)	x	0.25 up to 32µm
Automatic Weather Station	x	x
Particle Soot Absorption Photometer (EBC)	x	-
Silicon photodiode (UV-B)	x	x
Gas Chromotograph-MS analyser	x	-
Combined Relative humidity and Air temperature sensor	x	x
Silicon cell pyranometer	x	x
Integrating Nephelometer	x	x
Gas Chromotograph-ECD	x	-
Gas Chromotograph FID	x	-
Multi Angle Absorption	x	x
Gas Chromatograph RGD	x	-
Aerosol chemistry	x	x
Scanning Mobility Particle Size (15nm-32µm)	-	x

Figure A.2: List of atmospheric parameters for each station

Appendix B

WDBPALO Scripts

```
        #!/usr/bin/python
        import psycopg2
        import sys
        import string
        import pprint
        import wdbpaleo_connect as wdbp

        def opendata(datafile):
                lines = tuple(open(datafile, 'r'))
                #ins = open( datafile, "r" )
                #array = []
                #for line in ins:
            #           array.append( line )
                #ins.close()
                return lines

        def main():
                nomefile=raw_input('Inserisci il nome del file da elaborare:')
                dati=opendata(nomefile)
                #cursor=wdbp.connect()
                #execute our Query
                passw='sdbpaleo'#raw_input('Inserisci password sdbpaleo: ')
                conn_string = "host='localhost' dbname='sdbpaleo' user='sdbpaleo' password='%s'" %(passw)

                # print the connection string we will use to connect
                print "Connecting to database\n ->%s" % (conn_string)

                # get a connection, if a connect cannot be made an exception will be raised here
                conn = psycopg2.connect(conn_string)

                # conn.cursor will return a cursor object, you can use this cursor to perform queries
                cursor = conn.cursor()
                print "Connected!\n"
                line_one=dati[0]
            columns=0
            for i in range(0,len(line_one)-1):
                    a=raw_input
                    print line_one[i]
                    if string.find(line_one[i],';')!=-1: columns=columns+1
            print columns
            parameters=string.split(line_one,';')
            parameter=parameters[5:columns-1]
            print parameter
            el=string.split(dati[1],';')
            print el
            for i in range(1,len(dati)):

                    el=string.split(dati[i],';')
            #
            #       #print int(el[0]),int(el[1]),el[2],int(el[3]),int(el[4]),int(el[5]),int(el[6]),int(el[7]),
            #       #               el[8],el[9],el[10],el[11]
                    a1=string.lower(el[0]) #placename
                    a2=el[1] #identifier
                    a3=el[2] #section
                    a4=int(el[3]) #level_from
                    a5=int(el[4]) #level_to
                    a6='iamc.cnr' #dataprovidername
                    a7='depth' #levelparametername

                    print a1

                    cursor.execute("""SELECT * FROM wdb_int.coredatingmeasurement WHERE placename = \
                    lower(%(placename)s) AND (corerange).top<=%(levelfrom)s AND (corerange).bottom> \
                    =%(levelto)s""",{'placename':a1,'levelfrom':int(a4),'levelto':int(a5)})
                    date = cursor.fetchall()
                    pprint.pprint(date)
                    if date:
                        dato=date[0]
                        a8= float(dato[4]) #referencetime
                        a9=float(dato[5])  #validfrom
                        a10=float(dato[6]) #validto
                else:
                        dato=0
                        a8=-999.99
                        a9=-999.99
                        a10=-999.99
                for j in range (5,columns-1):
                        a11=parameter[j-5].replace("'","")
                        a12=float(el[j])
                        cursor.execute("select wci.begin('sdbpaleo',1111,1111,1111);")
                        cursor.execute("SELECT wci.writepaleo(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s);",\
                        (a12,a6,a1,a3,a2,a8,a9,a10,a11,a7,a4,a5,0))
                        conn.commit()
                        records = cursor.fetchall()
                        pprint.pprint(records)
                        print a12,' ',a6,' ',a1,' ',a3,' ',a2,' ',a8,' ',a9,' ',a10,' ', a11, ' ',a7, ' ', a4,' ',a5,0

if __name__ == "__main__":
        main()
```

FIGURE B.1: Script to load marine sediment core data into WDBPALEO: it uses Psycopg Python Library

```
#!/usr/bin/python
import psycopg2
import sys

def connect():
        #Define our connection string
        passw=raw_input('Inserisci password wdbpaleo: ')
        conn_string = "host='localhost' dbname='wdbpaleo' user='wdbpaleo' password='%s'" %(passw)

        # print the connection string we will use to connect
        print "Connecting to database\n ->%s" % (conn_string)

        # get a connection, if a connect cannot be made an exception will be raised here
        conn = psycopg2.connect(conn_string)

        # conn.cursor will return a cursor object, you can use this cursor to perform queries
        cursor = conn.cursor()
        print "Connected!\n"
        return cursor, conn
#if __name__ == "__main__":
#       main()
```

FIGURE B.2: Pyhton script to WDBPALEO connection: it uses Psycopg Python Library

```bash
#!bin/bash
echo " "
echo "---------------------------------------- "
echo " "
echo "           Nextdata import data           "
echo "             Version 0.10 Beta            "
echo "          We have to import data!         "
echo "                                          "
echo "-------daniele.strigaro@gmail.com---------"
echo " "
echo "---------------------------------------- "
echo "Digitare lo user per la funzione wci.begin:"
        read user
echo "Digitare la path in cui è presente il file csv(es.:/home/user/pathto.csv:)"
        read pathimport
echo "Database username:"
        read userdb
echo "Database password:"
        read passwrdb
echo "Indirizzo server del database:"
        read hostdb
echo "Porta dell'indirizzo server:"
        read portdb
echo "Nome del database:"
        read dbname

dataversion_=0
setconfidencecode_=0

while IFS=, read value_ dataprovidername_ placename_ referencetime_ validfrom_ validto_
valueparametername_
                levelparametername_ levelfrom_ levelto_

        do
                for i in $dbname
                        do
                                echo """
                                        SELECT wci.begin('$user');
                                        SELECT wci.writepaleo(
                                                '$value_',
                                                '$dataprovidername_',
                                                '$placename_',
                                                '$referencetime_',
                                                '$validfrom_',
                                                '$validto_',
                                                '$valueparametername_',
                                                '$levelparametername_',
                                                '$levelfrom_',
                                                '$levelto_',
                                                '$dataversion_',
                                                '$setconfidencecode_'
                                                );
                                """ | psql -h $hostdb -p $portdb -U $userdb -d $dbname
                        done
        echo $value_, $double, $dataprovidername_, $placename_, $referencetime_, $validfrom_,
$validto_, $valueparametername_, $levelparametername_, $levelfrom_, $levelto_

done < $pathimport
IFS=$OIFS
```

FIGURE B.3: Script to load ice core data into WDBPALEO: it uses Bash script

```
ALTER FUNCTION wci.writepaleo(dataprovider_ text, value_ double precision, icecorename_ text,
referencetime_ real, validfrom_ real, validto_ real, valueparameter_ text, levelparameter_ text,
levelfrom_ real, levelto_ real) OWNER TO daniele;

SET search_path = wci_int, pg_catalog;

--
-- Name: _pgis_fn_nn(public.geometry, double precision, integer, integer, character varying, character
varying, character varying, character varying); Type: FUNCTION; Schema: wci_int; Owner: daniele
--

CREATE FUNCTION _pgis_fn_nn(geom1 public.geometry, distguess double precision, numnn integer,
maxslices integer, lookupset character varying, swhere character varying, sgid2field character varying,
sgeom2field character varying) RETURNS SETOF pgis_nn
    LANGUAGE plpgsql STABLE
    AS $$
DECLARE
    strsql text;
    rec wci_int.pgis_nn;
    ncollected integer;
    it integer;
--NOTE: it: the iteration we are currently at
--start at the bounding box of the object (expand 0) and move up until it has collected more objects
than we need or it = maxslices whichever event happens first
BEGIN
        IF geom1 IS NULL THEN
                RETURN;
        END IF;
    ncollected := 0; it := 0;
    WHILE ncollected < numnn AND it <= maxslices LOOP
        strsql := 'SELECT currentit.' || sgid2field || ', st_distance(ref.geom, currentit.' ||
sgeom2field || ') as dist FROM ' || lookupset || '  as currentit, (SELECT geometry(''' || CAST(geom1 As
text) || ''') As geom) As ref WHERE ' || swhere || ' AND st_distance(ref.geom, currentit.' ||
sgeom2field || ') <= ' || CAST(distguess As varchar(200)) || ' AND st_expand(ref.geom, ' || CAST
(distguess*it/maxslices As varchar(100)) ||  ') && currentit.' || sgeom2field || ' AND
wci_int.expandoverlap_metric(ref.geom, currentit.' || sgeom2field || ', ' || CAST(distguess As varchar
(200)) || ', ' || CAST(maxslices As varchar(200)) || ') = ' || CAST(it As varchar(100)) || ' ORDER BY
st_distance(ref.geom, currentit.' || sgeom2field || ') LIMIT ' ||
        CAST((numnn - ncollected) As varchar(200));
        -- RAISE NOTICE 'sql: %', strsql;
        FOR rec in EXECUTE (strsql) LOOP
            IF ncollected < numnn THEN
                ncollected := ncollected + 1;
                RETURN NEXT rec;
            ELSE
                EXIT;
            END IF;
        END LOOP;
        it := it + 1;
    END LOOP;
END
```

Figure B.4: WDBPALEO-IDB: wci.writepaleo new function

```
CREATE TYPE corelength AS
(
        top integer,
        bottom integer
);

CREATE SEQUENCE coreid;

--Core Dating measurement

CREATE TABLE wdb_int.coredatingmeasurement (
    coredateid                          serial NOT NULL,
    typeid                              integer NOT NULL,
    placenamespaceid                    integer NOT NULL,
    placename                           character varying(255) NOT NULL,
    validtime                           real,
    datevalidfrom                       real,
    datevalidto                         real,
    corerange                           corelength,
    label                               character varying(255) NOT NULL,
    climaticevent                       character varying(255) NOT NULL,
    ecobiozone                          character varying(255) NOT NULL,
    note                                character varying(255) NOT NULL,
    placenameupdatetime                 timestamp with time zone NOT NULL default 'now',
    CHECK ( datevalidfrom <= datevalidto )
);

REVOKE ALL ON wdb_int.coredatingmeasurement FROM public;
GRANT ALL ON wdb_int.coredatingmeasurement TO wdb_admin;
ALTER TABLE ONLY wdb_int.coredatingmeasurement
    ADD CONSTRAINT coredatingmeasurement_pkey PRIMARY KEY (coredateid);

REVOKE ALL ON wdb_int.coredatingmeasurement FROM public;
GRANT ALL ON wdb_int.coredatingmeasurement TO wdb_admin;
```

Figure B.5: The new SDB table wdb_int.coredatingmeasurement

```
CREATE OR REPLACE FUNCTION wci.writepaleo(value_ double precision, dataprovidername_ character varying(255),
                                          placename_ character varying(255), section_ character varying(255),
                                          identifier_ character varying(255),  referencetime_ real, validfrom_ real,
                                          validto_ real, valueparametername_ character varying(255),
                                          levelparametername_ character varying(255), levelfrom_ real,
                                          levelto_ real,  dataversion_ integer)
  RETURNS void AS
$BODY$
DECLARE
        dataProviderId_                     int := wci_int.idfromdataprovider( dataProviderName_ );
        placeid_                            bigint := wci_int.getplaceidpaleo( placeName_);
        normalizedValueParameter_           character varying(255) := wci_int.normalizeParameter( valueParameterName_ );
        valueParameterId_                   integer := wci_int.getvalueparameterid( normalizedValueParameter_ );
        normalizedLevelParameter_           character varying(255) := wci_int.normalizeLevelParameter( levelParameterName_ );
        levelParameterId_                   integer := wci_int.getlevelparameterid( normalizedLevelParameter_ );
        currentVersion_                     integer := dataVersion_;
        confidenceCode_                     integer;
        valuegid_                           integer;

        dataId int;
BEGIN
        -- Determine dataversion
        IF (currentVersion_ IS NULL OR currentVersion_ < 0) THEN
                SELECT
                        max(dataversion) INTO currentVersion_
                FROM
                        wci_int.floatvalue_v v
                WHERE
                        v.dataproviderid = dataProviderId_ AND
                        v.referencetime = referencetime_ AND
                        v.placeid = placeid_ AND
                        v.valueparameterid = valueParameterId_ AND
                        v.levelparameterid = levelParameterId_ AND
                        v.levelFrom = levelFrom_ AND
                        v.levelTo = levelTo_ AND
                        v.validtimeFrom = validFrom_ AND
                        v.validtimeTo = validTo_;
                RAISE DEBUG 'WCI.WRITE.CurrentVersion: %', currentVersion_;
                IF currentVersion_ IS NULL THEN
                        currentVersion_ := 0;
                ELSE
                        currentVersion_ := currentVersion_ + 1;
                END IF;
        END IF;
        IF confidenceCode_ IS NULL THEN
                confidenceCode_ := 0;
        END IF;

        INSERT INTO wci_int.floatvalue_v
        (
                value,
                dataproviderid,
                dataprovidername,
                placeid,
                placegeometry,
                placeindeterminatecode,
                referencetime,
                validtimefrom,
                validtimeto,
                validtimeindeterminatecode,
                valueparameterid,
                valueparametername,
                valueunitname,
                levelparameterid,
                levelparametername,
                levelunitname,
                levelFrom,
                levelTo,
                levelindeterminatecode,
                dataversion,
                maxdataversion,
                confidencecode,
                valuestoretime,
                valueid,
                valuetype,
                section,
                identifier
        )
        VALUES
        (
                value_,
                dataproviderid_,
                dataprovidername_,
                placeid_,
                NULL,
                NULL,
                referencetime_,
                validfrom_,
                validto_,
                0,    -- Exact
                valueparameterId_,
                NULL, NULL,
                levelparameterId_,
                levelparametername_, NULL,
                levelfrom_,
                levelto_,
                0, --Exact
                currentVersion_,
                currentVersion_,
                confidenceCode_,
                'now',
                NULL,
                NULL,
                section_,
                identifier_
        );
        EXECUTE 'SELECT MAX(valuegroupid) from wdb_int.floatvalueitem' INTO valuegid_;

        INSERT INTO wdb_int.coreidentifiers
                (       valuegroupid,
                        identifier,
                        section
                )
        VALUES
                (valuegid_,
                identifier_,
                section_);
END
$BODY$
  LANGUAGE plpgsql VOLATILE SECURITY DEFINER
  COST 100;
ALTER FUNCTION wci.writepaleo(  value_ double precision, dataprovidername_ character varying(255),
                                placename_ character varying(255), section_ character varying(255),
                                identifier_ character varying(255),  referencetime_ real, validfrom_ real,
                                validto_ real, valueparametername_ character varying(255), levelparametername_
                                character varying(255), levelfrom_ real, levelto_ real,  dataversion_ integer)
OWNER TO filippo;
```

Figure B.6: The new WDBPALEO-SDB function wci.writepaleo

```sql
-- View: wci_int.floatvalue_v

CREATE OR REPLACE VIEW wci_int.floatvalue_v AS
 SELECT vli.value, dp.dataproviderid, dp.dataprovidername, dp.dataprovidernameleftset,
dp.dataprovidernamerightset, pl.placename, pl.placeid, pl.placegeometry, pl.placeindeterminatecode,
pl.originalsrid, vli.referencetime,
vli.referencetime + vlg.validtimefrom AS validtimefrom,
vli.referencetime + vlg.validtimeto AS validtimeto,
vlg.validtimeindeterminatecode, vlg.valueparameterid,
vp.parametername AS valueparametername,
vp.unitname AS valueunitname, vlg.levelparameterid,
lp.parametername AS levelparametername,
lp.unitname AS levelunitname, vlg.levelfrom, vlg.levelto, vlg.levelindeterminatecode, vlg.dataversion,
vli.maxdataversion, vli.confidencecode, vli.valuestoretime,
0::bigint AS valueid,
0 AS valuetype,
cri.identifier,
cri.section
   FROM wdb_int.coreidentifiers cri, wdb_int.floatvalueitem vli, wdb_int.floatvaluegroup vlg,
wci_int.dataprovider_mv dp, wci_int.placedefinition_mv pl, wci_int.parameter_mv vp,
wci_int.parameter_mv lp, wci_int.getsessiondata() s(dataprovidernamespaceid, placenamespaceid,
parameternamespaceid)
  WHERE vli.valuegroupid = vlg.valuegroupid AND dp.dataprovidernamespaceid = s.dataprovidernamespaceid
AND pl.placenamespaceid = s.placenamespaceid AND vp.parameternamespaceid = s.parameternamespaceid AND
lp.parameternamespaceid = s.parameternamespaceid AND vlg.dataproviderid = dp.dataproviderid AND
vlg.placeid = pl.placeid  AND vlg.valueparameterid = vp.parameterid AND vlg.levelparameterid =
lp.parameterid AND vli.valuegroupid=cri.valuegroupid;

ALTER TABLE wci_int.floatvalue_v
  OWNER TO filippo;
GRANT ALL ON TABLE wci_int.floatvalue_v TO filippo;
GRANT ALL ON TABLE wci_int.floatvalue_v TO wdb_admin;
GRANT SELECT ON TABLE wci_int.floatvalue_v TO wdb_read;
GRANT SELECT, INSERT ON TABLE wci_int.floatvalue_v TO wdb_write;


-- Rule: wci_internal_floatvalue_insert ON wci_int.floatvalue_v

-- DROP RULE wci_internal_floatvalue_insert ON wci_int.floatvalue_v;

--CREATE OR REPLACE RULE wci_internal_floatvalue_insert AS ON INSERT TO wci_int.floatvalue_v  DO
INSTEAD  SELECT wci.writepaleo(new.value::double precision, new.dataprovidername,new.placename,
new.section, new.identifier, new.referencetime,new.validtimefrom, new.validtimeto,
new.valueparametername, new.levelparametername,new.levelfrom, new.levelto,new.dataversion) AS
writepaleo

CREATE OR REPLACE RULE wci_internal_floatvalue_insert AS ON INSERT TO wci_int.floatvalue_v DO INSTEAD
SELECT wci_int.write(new.dataproviderid, new.placeid, new.referencetime, new.validtimefrom,
new.validtimeto, new.validtimeindeterminatecode, new.valueparameterid, new.levelparameterid,
new.levelfrom, new.levelto, new.levelindeterminatecode, new.dataversion, new.confidencecode,
new.value::double precision) AS writepaleo;
```

Figure B.7: The new WDBPALEO-SDB view wci_int.floatvalue_v

Bibliography

UNEP. Recent trends in melting glaciers and tropospheric temperatures over the himalayas and summer monsoon rainfall over india. *Technical Report. United Nations Environment Programme (UNEP)*, pages –, 2009.

G. Grabherr, M. Gottfried, and Pauli H. Gloria: A global observation research initiative in alpine environments. *Mountain Research and Development*, 20:190–191, 2000.

P. Lemke, J. Ren, Alley R.B., Allison I.and Carrasco J., Flato G., Fujii Y., Kaser G., Mote P., Thomas R.H., and Zhang T. Observations: Changes in snow and ice and frozen ground. in: Climate change 2007: The physical science basis. *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, pages –, 2007. URL http://ipcc-wg1.ucar.edu/wg1/wg1-report.html.

M. Beniston, H.F. Diaz, and R.S. Bradley. Climatic change at high elevation sites: an overview. *Climatic Change*, 36:233–251, 1997.

M. Legrand, S. Preunkert, Wagenbach D., and Fischer H. Seasonally resolved alpine and greenland ice core records of anthropogenic hcl emissions over the 20th century. *Journal of Geophysical Research Atmospheres*, 107(12):4139–, 2002.

W. S. B. Paterson and E. D. Waddington. Past precipitation rates derived from ice core measurements: Methods and data analysis. *Reviews of Geophysics*, 22(2):123–130, 1984. URL doi:10.1029/RG022i002p00123.

R. Delmas. Environmental information from ice cores. *Reviews of Geophysics*, 30(1): 1–21, 1992.

L.G. Thompson, T. Yao, Mosley-Thompson E., Davis M.E., Henderson K.A., and Lin P. N. A high-resolution millennial record of the south asian monsoon from himalayan ice cores. *Science*, (289):1916–1919, 2 2000.

F. Lirer, M. Sprovieri, Vallefuoco M., Ferraro L., Pelosi N., Giordano L., and Capotondi L. Planktonic foraminifera as bio-indicators for monitoring the climatic changes occurred during the last 2000 years in the se tyrrhenian sea. *Integrative Zoology*, pages –, 2013. URL doi:10.1111/1749-4877.12083.

W.F. Ruddiman, editor. *Earth's Climate Past and Future*. W.H.Freeman and amp and Co Ltd, New York, 2001.

W. Dansgaard, S. J. Johnsen, Clausen H. B., Dahl-Jensen D., Gundestrup N. S., Hammer C. U., Hvidberg C. S., Steffensen J. P., Sveinbjòrnsdottir A. E., Jouzel J., and Bond G. Evidence for general instability of past climate from a 250-ka ice-core record. *Nature*, (364):218–220, 1993.

G. Bond, W. Showers, Cheseby M., Lotti R., Almasi P., deMenocal P., Priore P., Cullen H., Hajdas I., and Bonani G. A pervasive millennial-scale cycle in north atlantic holocene and glacial climates. *Science*, 278:1257–1266, 1997.

N. J. Shackleton, M. A. Hall, and Vincent E. Phase relationships between millennial-scale events 64,000–24,000 years ago. *Paleoceanography*, 15:565–569, 2000.

P. A. Mayewski, E. E. Rohling, Stager J. C., Karlen W., Maasch K. A., Meeker L. D., Meyerson E. A., Gasse F., van Kreveld S., Holmgren K., Lee-Thorp J., Rosqvist G., Rack F., Staubwasser M., Schneider R. R., and Steig E. J. Holocene climate variability. *Quaternary Research*, 62:243–255, 2004.

A. Hayes, editor. *Late Quaternary palaeoclimatic and palaeoecological changes in the Mediterranean Sea*. University of Southampton - Faculty of Science -Department of Oceanography, New York, 1999.

F. Locci, M. T. Melis, F. Dessì, P. Stocchi, Akinde, V. Bones, P. Bonasoni, and E. Vuillermoz. Implementation of a webgis service platform for high mountain climate research: the share geonetwork project. *Geoscience Data Journal*, Article Accepted:–, 2014.

M. T. Melis, F. Dessì, Locci F., Bonasoni P., and Vuillermoz E. Share geonetwork: a web-service platform for environmental data sharing. *Proc. SPIE 8795, First International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2013)*, pages –, 2013. URL doi:10.1117/12.2027602.

OCG. *Open Geospatial Consortium*, pages –, 2012. URL http://www.opengeospatial.org/standards.

J. Tandy. Wmo core metadata profile version 1.2. *WMO*, pages –, 2010.

M. Pumphrey. Geoserver 2.2.x user manual. pages –, 2009. URL http://docs.geoserver.org/stable/en/user/.

A. Marinoni, P. Cristofanelli, Calzolari F., Roccato F., Bonafè U., and Bonasoni P. Continuous measurements of aerosol physical parameters at the mt. cimone gaw station (2165 m asl and italy). *Science of the total environment*, pages 241–251, 2008.

M. Citterio, G. Verza, Diolaiuti G., and Smiraglia C. The aws on the ablation tongue of forni glacier. *Technical Report. 10th Alpine Glaciology Meeting 2006*, pages –, 2006.

S. Siddiqui, L. Bharati, Pant M., Gurung P., Rakhal B., and Maharjan L.D. Nepal: Building climate resilience of watersheds in mountain eco-regions - climate change and vulnerability mapping in watersheds in middle and high mountains of nepal. *Technical Report. International Water Management Institute (IWMI)*, pages –, 2012.

P. Bonasoni, P. Laj, Marinoni A., Sprenger M., Angelini F., Arduini J., Bonafè U., Calzolari F., Colombo T., Decesari S., Biagio C.D., di Sarra A.G., Evangelisti F., Duchi R., Facchini M., Duchi R., Facchini M., Fuzzi S., Gobbi G.P., Maione M., Panday A., Roccato F., Sellegri K., Venzac H., Verza G., Villani P., Vuillermoz E., and Cristofanelli P. Atmospheric brown clouds in the himalayas: first two years of continuous observations at the nepal climate observatory pyramid (5079 m). *Atmospheric Chemistry and Physics*, pages 7515–7531, 2010a.

K. Ueno, H. Iida, Yabuki H., Seko K., Lhakupa G., Kayastha R., Pokhrel A., Shrestha M., Yasunari T., and Nakawo M. Establishment of the gen automatic weather station (aws) in khumbu region - nepal himalayas. *Bulletin of Glacier Research*, 14:13–22, 1996.

G. Tartari, G.P. Verza, and Bertolani L. Meteorological data at the pyramid laboratory limnology of high altitude lakes in the mt. everest region (nepal). *Verbania Pallanza : Consiglio nazionale delle ricerche and Istituto italiano di idrobiologia*, 57:–, 1998.

L. Bertolani, M. Bollasina, and G. Tartari. Recent biennial variability of meteorological features in the eastern highland himalayas. *Geophysical Research Letters*, 27:2185–, 2000.

K. Ueno, R. Kayastha, Chitrakar M., Bajracharya O.R., Pokhrel A.P., Fujinami H., Kadota T., Iida H., Manandhar D.P., Hattori M., Yasunari T., and Nakawo M. Meteorological observations during 1994-2000 at the automatic weather station (gen-aws) in khumbu region - nepal himalayas. *Bulletin of Glaciological Research*, 18:22–30, 2001.

M. Bollasina, L. Bertolani, and G. Tartari. Meteorological observations in the khumbu valley and nepal himalayas and 1994-1999. *Bulletin of Glaciological Research*, 19:1–11, 2002.

K. Ueno and A. Pokhrel. Intra-seasonal variation of surface air temperature in nepal. himalayas. *MAUSAM*, 53:281–288, 2002.

P. Bonasoniand, P. Laj, F. Angelini, Arduini J., Bonafè U., Calzolari F., Cristofanelli P., Decesari S., Facchini M., Fuzzi S., Gobbi G.P., Maione M., Marinoni A., Petzold A., Roccato F., Roger J., Sellegri K., Sprenger M. Venzac H., Verza G.P., Villani P., and Vuillermoz E. The abc-pyramid atmospheric research observatory in himalaya for aerosol and ozone and halocarbon measurements. *Science of the total environment*, pages 252–261, 2007.

P. Bonasoni, E. Vuillermoz, Lentini G., Toffolon R., Verza G.P., Listo L., Milanesi D., Flury B., Mari F., Bocci A., Cristofanelli P., Marinoni A., Duchi R., Smiraglia C., Diolaiuti G., Tartari G., Salerno F., and Provenzale A. Scientific research activities of ev-k2-cnr in the karakorum region in pakistan - expert report of share (stationa at high altitude for research on the environment). *Technical Report - Ev-K2-CNR Committee*, pages –, 2010b.

pages –. URL http://www.ceop-he.org.

I. Zahumensky. Guidelines on quality control procedures for data from automatic weather stations. *WMO Technical Report*, pages –, 2004.

Institute T.N.M. Wdb - weather and water database. *Institute T.N.M.*, pages –, 2012. URL http://wdb.met.no.

S. Sumathi and S. Esakkirajan, editors. *Fundamentals of relational database management system.* Springer Science and Buisness Media, New York, 2007.

F. Lirer. Wp 1.5 - paleoclimate data from marine sediments. *Nextdata Scientific Report for the reference period 01/01/2012-31/12/2012*, pages –, 2012.

J. Stroeve and A. Nolin. Comparison of modisand misr-derived surface albedo with in situ measurements in greenland. *Proceedings of EARSeL-LISSIG-Workshop Observing our Cryosphere from Space*, pages –, 2002.

J. Stroeve, J.E. Box, and T. Haran. Evaluation of the modis (mod10a1) daily snow albedo product over the greenland ice sheet. *Remote Sensing of Environment*, 105: 155–171, 2006.

S. Liang. Narrowband to broadband conversions of land surface albedo i algorithm. *Remote Sensing of Environment*, 76:213–238, 2000.

B. Pinty, M. Taberner, Haemmerle V. R., Paradise S. R., E. Vermote, and Verstraete M. M. Global-scale comparison of misr and modis land surface albedos. *Journal of Climate*, 24:732–749, 2011. URL dai:10.1175/2010JCLI3709.1.

D. Michel, R. Philipona, C. Rucksthul, R. Vogt, and L. Vuilleumier. Performance and uncertainty of cnr1 net radiometers during a one-year field comparison. *Journal of atmospheric and oceanic technology*, 25:–, 2007.

M. Pape and M. Vohland. A comparison of total shortwawe surface albedo retrievals from modis and tm data. *IAPRS*, XXXVIII-Part 7B:–, 2010.

G. A. Riggs, D. K. Hall, and V. V. Salomonson. Modis snow products user guide to collection 5. pages –, 2006. URL http://nsidc.org/data/docs/daac/modis_v5/dorothy_snow_doc.pdf.

S. Liang, A. H. Strahler, and Walthall C. Retrieval of land surface albedo from satellite observations: A simulation study. *Journal of Applied Meteorology*, 38:712–725, 1999.

S. Liang, C.J. Shuey, A.L. Russ, H. Fang, M. Chen, C.L. Walthall, C.S.T. Daughtry, and R. Hunt Jr. Narrowband to broadband conversions of land surface albedo: Ii. validation. *Remote Sensing of Environment*, 82:25–41, 2002.

S. Morin, Y. Lejeune, B. Lesaffre, J. M. Panel, D. Poncet, P. David, and M. Sudul. A 18-yr long (1993–2011) snow and meteorological dataset from a mid-altitude mountain site (col de porte, france, 1325 m alt.) for driving and snowpack models. *Earth Syst. Sci. Data Discuss.*, 5:29–45, 2012. URL doi:10.5194/essdd-5-29-2012.

T.J. Yasunari, Q. Tan, K.M. Lau, P. Bonasoni, A. Marinoni, P. Laj, M. Ménégoz, T. Takemura, and M. Chin. Estimated range of black carbon dry deposition and the related snow albedo reduction over himalayan glaciers during dry pre-monsoon periods. *Atmospheric Environment*, pages –, 2012. URL doi:10.1016/j.atmosenv.2012.03.031.

F. Salerno, G. Viviano, Thakuri S., Flury B., Maskey R.K., Khanal S.N., Bhuju D., Carrer M., Bhochhibhoya S., Melis M.T., Giannino F., Staiano A., Carteni F., Mazzoleni S., Cogo A., Sapkota A., Shrestha S., Pandey R.K., and Manfredi E.C. Energy, forest and indoor air pollution models for sagarmatha national park and buffer zone and nepal. *Mountain Research and Development (MRD)*, 30:113–126, 2010.

G. Nigrelli and A. Marino. Dbclim: A web-based and open source relational database for rainfall event studies. *Computers and Geosciences*, 48:337–339, 2012.

Kipp-Zonen. Kipp and zonen: Instruction manual cnr1 net radiometer. *Kipp and Zonen*, (2):–, 2009.

J. Liu, C. Schaaf, Strahler A., Jiao Z., Shuai Y., Zhang Q., Roman M, Augustine J.A., and Dutton E.G. Validation of moderate resolution imaging spectroradiometer (modis) albedo retrieval algorithm: Dependence of albedo on solar zenith angle. *Journal of Geophysical Research - Atmospheres*, 114:–, 2009.