ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/328215784

KML cookbook (early version)

Technical Report · October 2018

CITATIONS
0

READS
1,099

2 authors:

Anna Mironova
University of Oslo
8 PUBLICATIONS 13 CITATIONS
SEE PROFILE

Alexander Minakov
University of Oslo
45 PUBLICATIONS 361 CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project NOR-R-AM View project

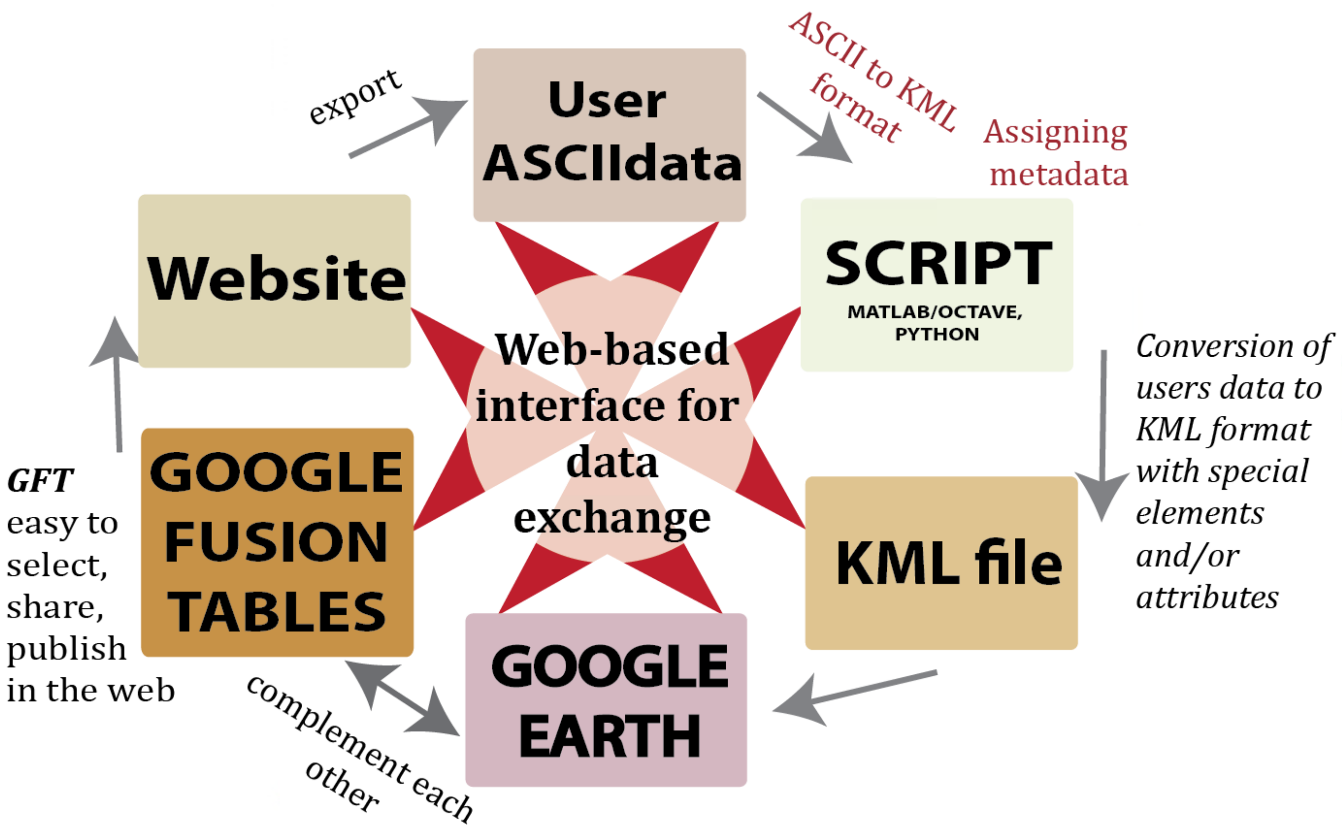
Project Transboundary problems associated with groundwater exploitation near the Russian-Estonian state boundary View project

All content following this page was uploaded by Alexander Minakov on 21 October 2018.

The user has requested enhancement of the downloaded file.

Cookbook for web-based interface for data exchange

Anna Mironova & Alexander Minakov


export
User ASCIIdata
ASCII to KML format
Assigning metadata
SCRIPT
MATLAB/OCTAVE, PYTHON
Conversion of users data to KML format with special elements and/or attributes
KML file
GOOGLE EARTH
complement each other
GOOGLE FUSION TABLES
GFT easy to select, share, publish in the web
Website
Web-based interface for data exchange

Centre for Earth Evolution and Dynamics,
University of Oslo, Norway
2018

Contents

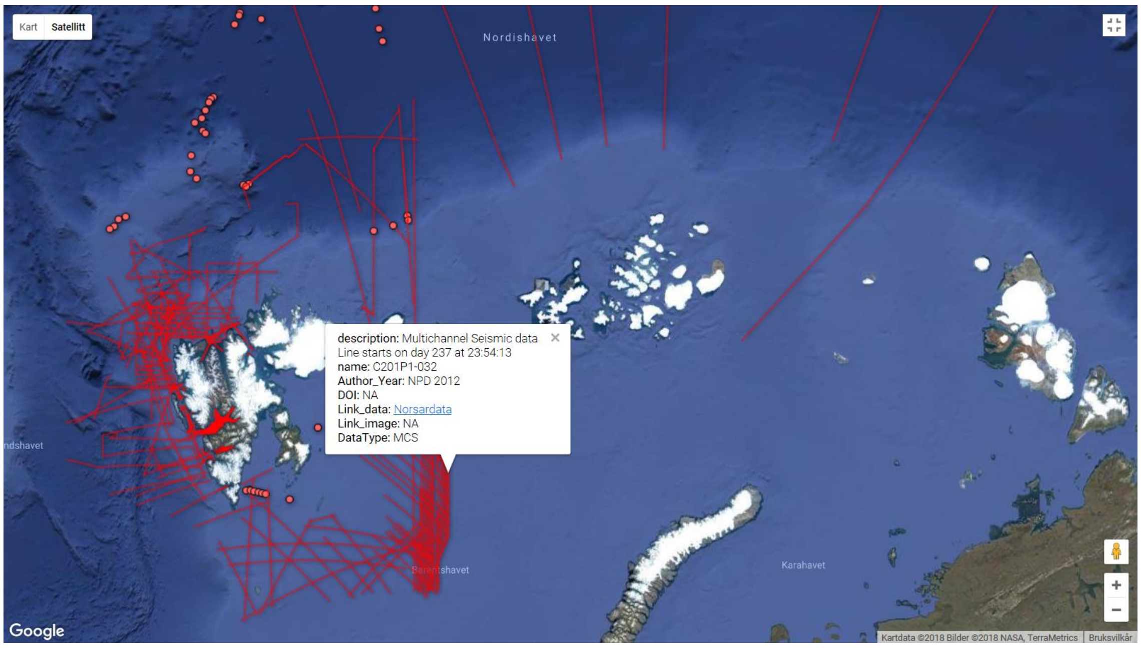
Contents ... 1
1. Preface ... 3
2. Type of users data ... 5
3. Keyhole Markup Language (KML) ... 5
3.1. Structure of KML files ... 5
3.2. Point ... 5
3.3. Lines ... 6
3.4. Adding attributes to KML files ... 6
3.5. Using Style and BalloonStyle elements as a template ... 7
3.6. Network Links ... 8
4. How to make a KML file? ... 8
4.1. Point data ... 9
4.1.1. Point data. Typical input data ... 9
4.1.2. Point data. Matlab script ... 9
4.1.3. Point data. Matlab script. Additional image in the description ... 10
4.1.4. Point data. Matlab script. Additional table in the description ... 11
4.1.5. Point data. Matlab script. Additional styles ... 11
4.1.6. Point data. Python script ... 12
4.1.7. Point data. Python script. Additional image in the description ... 13
4.1.8. Point data. Python script. Additional table in the description ... 13
4.1.9. Point data. Python script. Additional styles ... 13
4.1.10. Point data. Resulting KML file ... 14
4.2. Polyline data ... 15
4.2.1. Polyline data. Typical input data ... 15
4.2.2. Polyline data. Matlab script ... 15
4.2.3. Polyline data. Matlab script. Additional image in the description ... 16
4.2.4. Polyline data. Matlab script. Additional styles ... 16
4.2.5. Polyline data. Python script ... 17
4.2.6. Polyline data. Python script. Additional image in the description ... 18
4.2.7. Polyline data. Python script. Additional styles ... 18
4.2.8. Polyline data. Resulting KML file ... 19
4.3. Conversion KML file ... 20
4.3.1. From SHP to KML ... 20
4.3.2. From KML to GPML ... 20
5. Useful applications for work with database ... 20
5.1. Google Fusion Tables ... 20
5.1.1. Upload of FT on Google account ... 20
5.1.2. Import data ... 21
5.1.3. Data visualization on map ... 23
5.1.4. Sharing and publishing maps in website ... 26
5.2. Google Earth ... 27
5.2.1. Map ... 27
5.3. Anaconda distribution for Python ... 31
5.4. MATLAB/Octave ... 31
5.4.1.Comparing Matlab and Python scripts ... 31
6. Conclusion ... 32
Appendix ... 33

1. Preface

The database was created as a pilot project with financing from the first of the Go-North project, and then NOR-R-RAM. As part of an international scientific networking program, it was building a web database for the Arctic region, where published geophysical data are easily visualized and retrieved. The database was created by Anna Mironova, under the conceptual leadership of Alexander Minakov and the support of Carmen Gaina and Jan Inge Faleide.
Why do we create a web-database?
We created a web-database built on open resources and accessible to all users via Google services:

- Independent of commercial cartographic resources
- Includes only published and available information
- Interactive for changing and completing
- Easy-to-use and fill with new data
- Filled with a large number of geophysical information about the Arctic
- The file format easily compatible with commercial GIS
- Possibility of supplementing with monitoring data from geophysical institutions using the same KML format, such as NORSAR, IODP (International Ocean Drilling Program), Oceanographic institution of California and another.

It has been recently found an easy-to-implement solution suitable for geoscience-related spatial databases. Google application Fusion tables has been used for visualization of geophysical data in the interactive map with opportunity to share information with another researchers groups and presenting it on the website (Fig.1).


Kart
Satellitt
Nordishavet
ndshavet
Barentshavet
Karahavet
Google
Kartdata ©2018 Bilder ©2018 NASA, TerraMetrics | Bruksvilkår

description: Multichannel Seismic data
Line starts on day 237 at 23:54:13
name: C201P1-032
Author_Year: NPD 2012
DOI: NA
Link_data: Norsardata
Link_image: NA
DataType: MCS

Figure 1. Map view of database on web site https://norramarctic.wordpress.com/resources-and-links/ with lines and point data. In the balloon window are shown seismic metadata information.

Than we click to the drawing objects on the map in Google Earth, Fusion tables or website, than appeared on the screen balloon window with metadata information from KMK file.

Database consist of a set of KML-files including link to actual geophysical data such as:

- Multichannel seismic (MCS) profiles/image (462 profiles)
- Seismic refraction (OBS, Sonobuoys hydroacoustic data – 188 stations) profiles/crustal thickness, velocities
- Permanent/temporal seismic stations

- Potential fields/Grids
- Heat flow measurements
- Magnetic data and interpretations (isochrones/picks)
- Present plate boundaries and other line geometries

The concept of the web-database (Fig.2) is that we fill it with various content, and then import the data to Google Earth or to the Google Fusion Table from which we can share and upload data for further analysis.

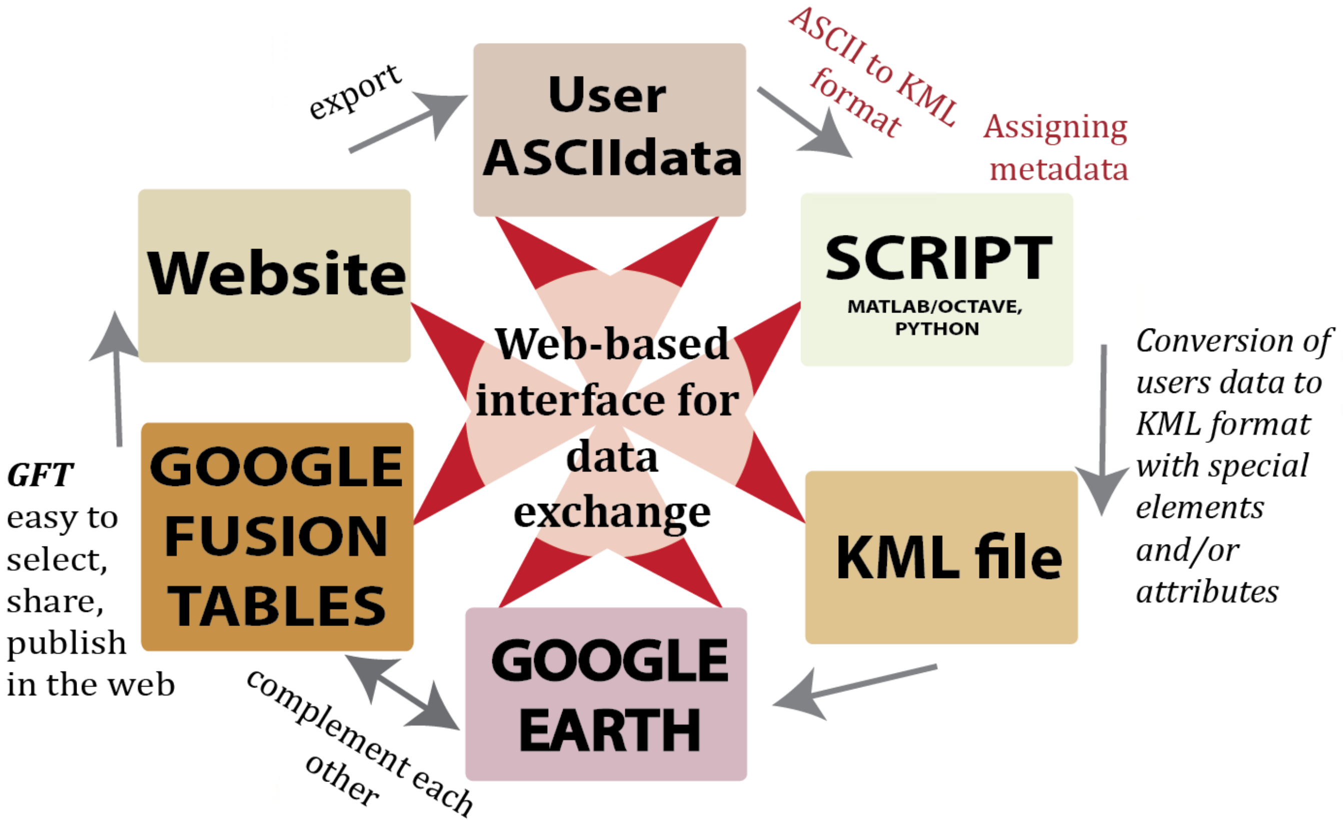

export
User ASCIIdata
ASCII to KML format
Assigning metadata
Website
SCRIPT
MATLAB/OCTAVE, PYTHON
Web-based interface for data exchange
Conversion of users data to KML format with special elements and/or attributes
GFT easy to select, share, publish in the web
GOOGLE FUSION TABLES
KML file
GOOGLE EARTH
complement each other

Figure 2. Flow chart of concept web-database.

Input data can be raw data or final modeling results, therefore both vector and raster data (e.g. through WMS/WCS services), as well as detailed metadata. For example, the desired azimuth, distance and time period of the seismic sources from air-gun shots can be efficiently selected using interactive maps.
Simple scripts convert input geographical coordinates data by adding metadata, geometry in the form of KML format, and retrieve the geolocation of data on map in readable form when required. Result presented in Fig.1.

This mode of view can be useful for:

- Seismic analysis
- Creating of 3D velocity models
-
-
-
-
-

We would like to hear any suggestions for future enhancements and modification of database. Please send your comments to Alexander Minakov.

2. Type of users data

This cookbook explains how to upload information to web site and work with Web database, Fusion tables, make KML files based on geophysical data and adding metadata to its. Likewise, you will get here different geophysical data types like earthquake seismology, MCS data, sonobuoys and other.

Data from different format have been translated into the simple text format txt. All working data files are transferred to the electronic view and are either on the owner's server or copied to a virtual cloud on Dropbox server corresponding to Alexander Minakov university-business account: *Go-North*. All database in text format link. Fusion table database *link*.

The waveform data, corresponding to the air-gun shot times (UKOOA files), was extracted from the IRIS and other FDSN servers. Routines are provided to access the events information, station metadata, and time series data. Thus, we get the information about the stations (coordinates, channel, elevation, sensitivity, and sampling rate), shot time and location, and three-component waveform data. We linked the seismological data via the free Google Earth browser and analyzed it together with other types of geological/geophysical information. We showed some examples of data processing (including band-pass filtering, time-frequency analysis, phase rotation, picking, stacking) and simple travel time modeling to infer the seismic velocity structure. It was integrated for a more quantitative analysis using the Google Earth browser and simple processing scripts.

3. Keyhole Markup Language (KML)

Keyhole Markup Language (KML) is an XML notation for expressing geographic annotation and visualization within Internet-based and map. KML was developed for use with Google Earth, which was originally named Keyhole Earth Viewer. KML is an international standard maintained by the Open Geospatial Consortium, Inc. (OGC). If you're new to KML, begin by used the *KML Tutorial*, which presents short samples of KML code that you can view in Google Earth.

3.1. Structure of KML files

The KML file specifies a set of features (place marks, images, polygons, models, textual descriptions, etc.) that can be displayed on maps in geospatial software implementing the KML encoding. Each place always has a longitude and a latitude.
The structure of KML file for lines, points and overlay images are similar and breaks down as follows:

- An XML header. This is line 1 in every KML file. No spaces or other characters can appear before this line.
- A KML namespace declaration. This is line 2 in every KML file.

A ***Placemark*** object that contains the following elements:

- A ***name*** that is used as the label for the ***Placemark***
- A ***description*** that appears in the "balloon" attached to the ***Placemark***

The ***description*** can include links, font sizes, styles, colors, and specify text alignment and tables.

3.2. Point

A ***Placemark*** is a mark of position on the Earth's surface, shown for example, as a yellow pushpin. The simplest ***Placemark*** includes only a <***Point***> element, which specifies the location of the ***Placemark***.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">

 <Placemark>
   <name> sonobuoy 1</name>
   <description> Nansen Basin sonobuoy data. </description>
   <Point>
     <coordinates>122.08, 37.42 , 0</coordinates>
   </Point>
  </Placemark>

</kml>
```

The feature of structure for this file is:

- A ***Point*** that specifies the position of the ***Placemark*** on the Earth's surface with *coordinates* (longitude, latitude, and optional altitude)

3.3. Lines

In KML, a path (polyline) is created by a <**LineString**> element. ***LineString*** that specifies the position of the ***Placemark*** on the Earth's surface with *coordinates* (longitude, latitude, and altitude for each point of line)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">

  <Document>
    <name>ARC14 </name>
    <description> Barents sea </description>¨
 <Placemark>
     <name>ARC14-039 </name>
     <description> Multichannel Seismic data </description>
     <LineString>
       <coordinates>
                   15.6675, 89.0211, 0
                   15.6675, 90, 0
                   -164.0808, 90, 0
                   -164.0808, 89.2058, 0
                   -180, 89.1674, 0
                   180, 89.1674, 0
                   171.4243, 89.11, 0
                   172.0596 ,88.6319, 0
                   163.4784, 85.9732, 0
                   165.4885, 84.6071, 0
                   170.0953, 82.3502, 0
       </coordinates>
     </LineString>
    </Placemark>
  </Document>
</kml>
```

3.4. Adding attributes to KML files

For the database, we added specified attributes to KML files, that could help us in the next step to unify loading information, facilitate the process of sorting and searching for the required data in for example Fusion tables. In KML we used for it <**ExtendedData**> and specified style <**BalloonStyle**>. To make the upload information the same for all users, we have selected the following key items to be displayed in the KML file:

- Author and year of publication,
- Bibliographic codes/doi of publications,
- Include links to the data repositories and images,

- Data type.

The Keyhole Markup Language offers to add key items data to a KML Feature. The <**ExtendedData**> element provides the following mechanisms:
<Data> element - allows to add untyped name/value pairs to the user data associated with a given Feature (author and year of publication, bibliographic codes/doi for publications, and include links to the data repositories and images, data type. These name/value pairs are displayed in the balloon by default. This information can also be used for entity replacement in the <text> element of <**BalloonStyle**>.

```
<ExtendedData>
		<Data name="Author_Year">
				<value>  Nikishin A.M. et al. 2017  </value> </Data>
		<Data name="DOI">
				<value>  <![CDATA[<href>  <  a  href="https://www.sciencedirect.com/science/article/pii/S0040195117303669?via%3Dihub">  DOI
</a></href>]]> </value> </Data>
		<Data name="Link_data">
				<value> <![CDATA[<href>  NA  </href>]]></value> </Data>
		<Data name="Link_image">
				<value>  <![CDATA[<  a  href="https://www.dropbox.com/s/bbimoqwtrr1b7ob/Arc14-39b.pdf?dl=0">  Hight  resolution  image
</a>]]></value> </Data>
		<Data name="DataType">
				<value>  MCS  </value> </Data>
</ExtendedData>
```

3.5. Using Style and BalloonStyle elements as a template

In KML, is it possible to define a Style once and assign an ID to it. After defining the Style in this manner, is it easy to reference it multiple times within the KML file using the <styleUrl> element. A style defined in this way is referred to as a shared style. The <text> element within <**BalloonStyle**> supports entity replacement. Individual values can be substituted for each instance of the entity. Geophysical community can approved KML element for best visualization on Fusion tables. Entities that we replaced and decided for web-database elements are as follows:

$[name]	Replaced with the name of the object
$[description]	Replaced with the description of the Object
$[Author_Year]	Replaced with the Author of article or report of the Object
$[DOI]	Replaced with the DOI - bibliographic codes of article or report of the Object
$[Link_data]	Replaced with the links to the data repositories of article or report of the Object
$[Link_image]	Replaced with the links to the image repositories of article or report of the Object
$[DataType]	Replaced with the type of data of the Object

Here is an example that creates a <**BalloonStyle**> template. For each object in balloon the Google Earth substitutes the name of the object and then writes out information containing in the <**ExtendedData**>.

```
<Style id="style">
		<BalloonStyle>
				<text><![CDATA[<Center><B>$[name]</B>
				</Center><br>$[description]<P><br/>
				Author: $[Author_Year]<br/>
				DOI: $[DOI]<br/>
				Link to data: $[Link_data]<br>
				Link to image: $[Link_image]<br/>
				Data type: $[DataType]]]></text>
		</BalloonStyle>
		<LineStyle>
				<color>80ff69b4</color>
				<width>2</width>
		</LineStyle>
	</Style>
```

3.6. Network Links

A network link contains a <Link> element with an <href> (a hypertext reference) that loads a file. The <href> can be a local file specification or an absolute URL. Despite the name, a <NetworkLink> does not necessarily load files from the network.

The <href> in a link specifies the location of any of the following:

- An image file used by icons in icon styles, ground overlays, and screen overlays
- A model file used in the <Model> element
- A KML or KMZ file loaded by a Network Link

The specified file can be either a local file or a file on a remote server. In their simplest form, network links are a useful way to split one large KML file into smaller, more manageable files on the same computer.

Live links inside KML files were assigned to:

- Article (DOI)
- Data archive
- Image

We used 'NA' than we had not information or than link did not exist.
We have chosen three main storages for links:

- Institute servers http://www.uio.no/ (UIO)
- B2Drop https://eudat.eu/ (European Open Science Cloud)
- Dropbox (Cloud File, Sharing and Storage)

It was find optimal way to present images in the database.
For <**description**> image can be seen on the screen in a pop up balloon. Example link is presented below:

```
<img src="https://dl.dropboxusercontent.com/s/ bbimoqwtrr1b7ob/Arc14-39b.pdf?dl=0" width=500>.
```

However for <**BalloonStyle**> was choosed to seen not whole image but just link to image from the same web storage with some comment like here "High resolution image", example:

```
<a href="https://www.dropbox.com/s/bbimoqwtrr1b7ob/Arc14-39b.pdf?dl=0"> High resolution image </a >
```

As you can see the above two links are similar. The beginning part of the links, above highlighted with red color, are different from second part of links, highlighted with blue. It supposed us to play with visualization of image in web-database. Links are identical after first part.
Place of image size is written behind quotes. Image size in pixels regulates with parameter of <width=> or combination of <width = > and <high = >.

4. How to make a KML file?

The simplest kind of KML documents are those that can be authored directly in Google Earth – it does not need to edit or create any KML in a text editor. Placemarks, paths, and image overlay can all be make directly in Google Earth (GE).
We can present in the GE map many data like Placemark (point), Path (lines), and Image Overlay (raster or WMS). Simple placemarks, paths, and image overlay can all be created directly in Google Earth (Fig. 3) and saved like KML files.

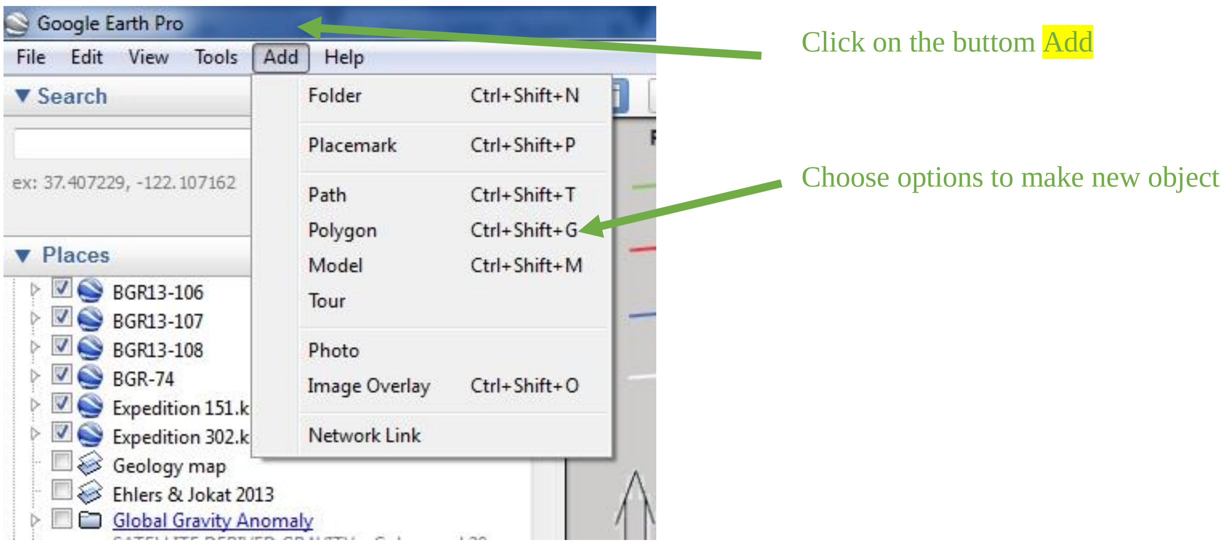


Figure 3. Screenshot from Google Earth program, example describes making simple KML inside GE.

Typing thousands or millions of elements could be problematic for geophysical data such as multichannel seismic or hundreds of sonobuoys point's data, so we propose to generate scripts in Matlab or Python to present information on the map in KML format. This cookbook contains sample codes that show how to generate components of database in KML. These examples below show how to load points and polylines from a text file and assign to web database necessary attributes.

4.1. Point data

4.1.1. Point data. Typical input data

An example input text file for sonobuoys geophysical measurements (number of sonobuoy/ latitude / longitude / MAX offset (m)/ Water depth / Velocity (top) / Velocity (bottom)) looks like:

```
1     81.65   4.31    7.3     0.770   1.70    1.70
2     81.68   4.77    12.6    0.760   1.61    1.61
3     81.78   5.20    11.3    0.790   1.82    1.82
4     81.83   5.93    14.6    0.812   1.88    1.88
6     82.34   12.94   12.4    1.047   1.60    1.60
7     82.44   12.33   14.8    0.860   3.88    999
9     82.64   12.38   2.5     1.320   999     999
10    82.92   13.82   5.3     3.270   1.70    1.70
```

Typical example of input file for point data. Here can see final 2-D Modelled Sonobuoy Solutions of the Nansen Basin, Arctic Ocean from Engen et al. File is presented in the folder with Examples: Data_point.txt.

At the first step, we using for the first three columns with coordinates. Last columns we will used like additional information inside the description for each sonobuoys point measurements with table views of information.

4.1.2. Point data. Matlab script

This example describe:

- uploads some point locations from a text file,
- writes the parameters of point style (like color and size) to a KML file,
- sets a **description** with images and links.

We have marked specifications and comments inside scripts below with the green color and lattice #. Input file Data_point.txt and output file M_point.kml you can find it in Examples folder.

```
clear
%%
```

```
% Generate lines for new KML file with name: M_point.kml
filename = 'M_point.kml';
%Set new name for sonobuoys data
name = 'Nansen Basin sonobuoy data';
%Open a file in write mode.
fid = fopen(filename, 'wt');
% Generate header for KML
header = ['<?xml version="1.0" encoding="UTF-8"?>',10,...
    '<kml xmlns="http://earth.google.com/kml/2.1">',10,...
    '<Document>',10,...
    '<name>',10,name,10,'</name>',10,...
    ];
% Generate style of sonobuoy point (link to the icon image)
mystyle =  [ '<Style id="style">', 10,...
          '<IconStyle>', 10,'<Icon>', 10,...
          '<href> http://maps.google.com/mapfiles/kml/pushpin/ylw-pushpin.png</href>', 10,...
          '</Icon>', 10,'</IconStyle>', 10,...
          '</Style>', 10];

%Load data from txt file with sonobuoys information
xy_coo = load('..\Data_point.txt');
% Create a selection of unique values of the sonobuoy's number
obsnum = fix(xy_coo(:,1));
% Assign the values of sonobuoys coordinates like latitude, longitude and elevation
lat =  xy_coo(:,2);
lon =  xy_coo(:,3);
el  = xy_coo(:,4);
geoData = [];

%Loop of elements from the first until the size of data table.
for i = 1:size(xy_coo,1)
% Assign the attributes of KML file. #10 means new line in KML. Name of sonobuoy converted into
string format with fixed name for each point. Description of point with possible images from
modeling. Style of point. Points coordinates converted into string format for each point
    KMLstring = ['<Placemark>',10,...
        '<name>',['Nb_SB',num2str(fix(obsnum(i)))],'</name>',10,...
        '<description>',10,...
        ['<![CDATA[','Seismic stratigraphy and sediment thickness of the Nansen Basin, Arctic
Ocean','<br>',10,...
       ']]>'],10,...
        '</description>',10,...
       '<styleUrl>#style</styleUrl>',10,...
     '<Point>',10,...
            '<coordinates>',10,...
            num2str(lon(i)),',',num2str(lat(i)),',',num2str(-el(i)),10,...
        '</coordinates>',10,...
     '</Point>',10,...
    '</Placemark>'
    ];
geoData = [geoData,KMLstring];
end
% Generate footer for KML
footer = [10,'</Document>',10,...
    '</kml>'];
% Finally write the list of header, style, sonobuoy points, footer into KML file
fprintf(fid,'%s',header);
fprintf(fid,'%s',mystyle);
fprintf(fid,'%s',geoData);
fprintf(fid,'%s',footer);
fclose(fid);
```

File with MATLAB script is presented in the folder with Examples: M_point.m, resulting KML file is M_point.kml.

4.1.3. Point data. Matlab script. Additional image in the description

Assigns to KML points file with tag `<description>` a description with images from local computer or link's location from cloud. Input file is Data_point.txt, output file is M_point_image.kml from Examples folder.

```
# Description of point with possible images from modeling
'<description>',10,...
       ['<![CDATA[','Sonobuoy, Nansen Basin, Arctic Ocean', '<br>',10,...
       '<img src="https://dl.dropboxusercontent.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0"
width=400>','<br>',10,...
```

```
        ']]>'],10,...
'</description>',10,...
```
File with complete MATLAB script is presented in the folder with Examples: M_point_image.m, resulting KML file is M_point_image.kml.

4.1.4. Point data. Matlab script. Additional table in the description

Assigns to KML points file with tag `<description>` a table from Data_point.txt file

```
'<description>',10,...
        ['<![CDATA[','Sonobuoy, Nansen Basin, Arctic Ocean',' <br>',10,...
        '<table border="1" padding="1">',10,...
        '<tr><th>MAX offset(m)</th><th> WD </th><th>V(top)</th><th>V(bot)</th>',10,...

'<tr><td>',num2str(xy_coo(i,4)),'</td><td>',num2str(xy_coo(i,5)),'</td><td>',num2str(xy_coo(i,6)),'</
td><td>',num2str(xy_coo(i,7)),'</td></tr>',10,...
    '</table>',10,...
    '<br> SB - sonobuoy number; WD - water depth. Depths in km, velocities in km s-1.',10,...
    ']]>'],10,...
'</description>',10,...
```
File with complete MATLAB script is presented in the folder with Examples: M_point_all.m, resulting KML file is M_point_all.kml.

4.1.5. Point data. Matlab script. Additional styles

Assigns the community approved KML elements attributes to each point, which should be appeared on the map with click on the sonobuoy point in a balloon. Input file is Data_point.txt, output file is M_point_all.kml from Examples folder.

```
% Assign the community approved KML elements values for all sonobuoys
authorList = 'Engen et al. 2009';
bibLinks   = '<a href="https://doi.org/10.1111/j.1365-246X.2008.04028.x">DOI</a>';
dataLink = 'NA'; %'NA' if does not exist any information
imgLink = '<a href="https://www.dropbox.com/s/... ">Image</a>';
% Extended Data with attributes are the same for all database objects and different content
% Generate style of sonobuoy points and style inside the balloon window near points, that appearing
with click on point
mystyle =  [ '<Style id="style">',10,...
          '<IconStyle>',10,'<Icon>',10,...
          '<href>http://maps.google.com/mapfiles/kml/pal3/icon19.png </href>',10,...
          '</Icon>',10,'</IconStyle>',10,...
          '<BalloonStyle>',10,...
            '<text>',10,...
                '<![CDATA[<B>$[name]</B><br>$[description]<P><br/>',10,...
'Author: $[Author_Year]<br/>',10,...
'DOI: $[DOI]<br/>',10,...
'Link to data: $[Link_data]<br>',10,...
'Link to image: $[Link_image]<br>',10,...
'Data type: $[DataType]]]>',10,...
            '</text>',10,...
  '</BalloonStyle>',10,'</Style>',10,...
  ];
% Extended Data with attributes are the same for all database objects and different content
      '<ExtendedData>',10,...
       '<Data name="Author_Year">',10,...
               '<value>',authorList,'</value>',10,...
       '</Data>',10,...
       '<Data name="DOI">',10,...
               '<value>',bibLinks,'</value>',10,...
       '</Data>',10,...
       '<Data name="Link_data">',10,...
               '<value>',dataLink,'</value>',10,...
        '</Data>',10,...
        '<Data name="Link_image">',10,...
               '<value>',imgLink,'</value>',10,...
        '</Data>',10,...
        '<Data name="DataType">',10,...
               '<value>','SB','</value>',10,...
        '</Data>',10,...
      '</ExtendedData>',10,...
```

File with complete MATLAB script is presented in the examples folder: M_point_all.m, resulting KML file is M_point_all.kml.

4.1.6. Point data. Python script

For programing in Python language, we used Anaconda navigator with Spyder 3.2.6. open source software. It is the scientific interactive development environment for the Python language with advanced editing, interactive testing and introspection features and a numerical computing environment. Spyder helpful for scientific computing, its interface is easy to use for Matlab users.

Python language cannot read character “N”, “n”, “\N”, “\n” in file name or inside script body. Python reserved this symbol for action in programming. Be careful, did not use it without reason.
This example describe:

- uploads some point locations from a text file,
- writes the parameters of point style (like color and size) to a KML file,
- sets a **description** with images and links.

Input file is Data_point.txt, output file is P_point.kml and you can fide it in Examples folder.

```
# Automatically added attributes for Python file
# -*- coding: utf-8 -*-
"""

# Generate header for KML
header = "<?xml version=\"1.0\" encoding=\"UTF-8\"?> \n \
    <kml xmlns=\"http://earth.google.com/kml/2.1\"> \n \
    <Document> \n \
    <name> \n Sonabuoy Nansens basin \n </name> \n"
# Generate new KML file and feed it
f = open("P_point.kml", "w")

mystyle = <Style id="s_ylw-pushpin_hl"> \n \
<IconStyle> \n <scale>1.2</scale> <Icon> \n \
<href> http://maps.google.com/mapfiles/kml/pushpin/ylw-pushpin.png </href> \n \
</Icon> \n </IconStyle> \n \
</Style> \n"

#Upload NumPy library for Python programming language
import numpy as np
#Open file and make array of float numbers
subtable = np.genfromtxt("..\Data_point.txt", delimiter = '', dtype = np.float)
#Find the size of number of elements in subtable
nz = np.size(subtable,0)
# Make array with number
subtable = np.array(subtable)
# Assign header and style for KML
f.write(str(header))
f.write(str(mystyle))

# Create a selection of unique values of the sonobuoy’s number from the subtable and loop of elements
from the first until the size of data subtable.
for i in range(nz):
# Description of point from sonobuoys modeling
    descr = "<![CDATA[Seismic stratigraphy and sediment thickness of the Nansen Basin, Arctic Ocean
<br></br> \n \ ]]>"
# Assign the attributes of KML file name, description.
    f.write(("<Placemark>\n") + \
    ("<name> SB " + str(int(subtable[i,0])) + "</name> \n")+ \
    ("<description>"+"\n" + descr + "</description>" + "\n")+ \
    ("<styleUrl> #style </styleUrl> \n")+ \
    ("\t\t<Point>\n")+ \
    ("\t\t\t<coordinates>" + str(subtable[i,2]) + " ," + str(subtable[i,1]) + ", " +
str(subtable[i,3]) + "</coordinates>" + "\n")+ \
    ("\t\t</Point>\n")+ \
    ("</Placemark>\n"))
f.write(("</Document>\n")+ \
("</kml>\n"))
f.close()
```

File with Python script is presented in the folder with Examples: P_point.py, resulting KML file is P_point.kml.

4.1.7. Point data. Python script. Additional image in the description

Assigns to KML points file with tag `<description>` a description with images from local computer or link's location from cloud. Input file is Data_point.txt, output file is P_point_image.kml from Examples folder.

```
# Description of point with images from modeling
    descr = "<![CDATA[Sonobuoy, Nansen Basin, Arctic Ocean <br></br> \
    <Center><img        src=\"https://dl.dropboxusercontent.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0\"
width=400></Center>]]>"
```

File with complete Python script is presented in the folder with Examples: P_point_image.py, resulting KML file is P_point_image.kml.

4.1.8. Point data. Python script. Additional table in the description

Assigns to KML points file with tag `<description>` a table from the same Data_point.txt file

```
# Description of point with table
descr  =  "<![CDATA[Seismic  stratigraphy  and  sediment  thickness  of  the  Nansen  Basin,  Arctic  Ocean
<br></br> \n \
    <table border=\"1\" padding=\"1\"> \n\
    <tr><th>MAX offset(m)</th><th>WD</th><th>V(top)</th><th>V(bot)</th> \n\
    <tr><td>"  +    str(subtable[i,  3])  +  "</td><td>"  +    str(subtable[i,4])  +  "</td><td>"  +
str(subtable[i,5]) + "</td><td>" +  str(subtable[i,6]) + "</td></tr>\n \
    </table>\n \
    SB - sonobuoy number; WD - water depth. Depths in km, velocities in km s-1.   \n\
    ]]>"
```

File with complete Python script is presented in the folder with Examples: P_point_all.py, resulting KML file is P_point_all.kml.

4.1.9. Point data. Python script. Additional styles

Assigns attributes to each point in KML file, which could be present inside a balloon on the map, appearing with click on location of sonobuoy. Input file Data_point, output file P_point_all.kml.

```
# Assign the community approved KML elements values for sonobuoys
authorList = 'Engen et al. 2009'
bibLinks = '<a href="https://doi.org/10.1111/j.1365-246X.2008.04028.x ">DOI</a>'
dataLink = 'NA'
imgLink = '<a href="https://www.dropbox.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0">Image</a>'

# Generate style of sonobuoy icon, label and style of bubble window appearing near point with click
mystyle = "<Style id=\"style\"> \n \
<IconStyle> \n <color>9014F0F0</color> \n <scale>1.2</scale> \n \
<Icon> \n <href>http://maps.google.com/mapfiles/kml/shapes/donut.png </href> \n
</Icon> \n </IconStyle> \n \
<LabelStyle><scale>0.1</scale></LabelStyle> \n \
<BalloonStyle> \n <text> \n \
<![CDATA[<Center><B>$[name]</Center><br><br>$[description]</B><P><br> \
Author: $[Author_Year]<br/> \
DOI: $[DOI]<br> \
Link to data: $[Link_data]<br>  \
Link to image: $[Link_image]<br>  \
Data type: $[DataType]<br>]]> \n </text> \n \
</BalloonStyle> \n </Style> \n"

# Assign the attributes of extended data.
f.write(("\t\t<ExtendedData> \n")+ \
    ("\t\t<Data name=\t\"Author_Year\"> \n" + "<value>" + authorList + "</value>\n </Data>\n")+ \
    ("\t\t<Data name=\"DOI\">\n <value>" + bibLinks + "</value> \n </Data> \n")+ \
    ("\t\t<Data name=\"Link_data\"> \n <value>" + dataLink + "</value>\n </Data> \n")+ \
    ("\t\t<Data name=\"Link_image\"> \n <value>"+ imgLink + "</value>\n </Data> \n")+ \
    ("\t\t<Data name=\"DataType\"> \n <value> SB </value> \n </Data>\n ")+ \
    ("\t\t</ExtendedData> \n"))
```

File with complete Python script is presented in the folder with Examples: P_point_all.py, resulting KML file is P_point_all.kml.

4.1.10. Point data. Resulting KML file

At the result of addicting of complete metadata with help of Matlab (M_point_all.m) or Python (P_point_all.py) we have got KML file for sonobuoys (points) with name, description, image, table, author and year of publication, DOI, links to depository, type of data. Below we present resulting short version of KML point file with the first and second sonobuoy. Se in Example folder files M_point_all.kml and P_point_all.kml complete versions of KML files without reduction.

```
<?xml version="1.0" encoding="UTF-8"?>
     <kml xmlns="http://earth.google.com/kml/2.1">
     <Document>
     <name>  Sonabuoy Nansens basin  </name>
<Style id="style">
 <IconStyle>
 <color>9014F0F0</color>
 <scale>1.2</scale>
 <Icon>
 <href>http://maps.google.com/mapfiles/kml/shapes/donut.png</href>
</Icon>
 </IconStyle>
 <LabelStyle><scale>0.1</scale></LabelStyle>
 <BalloonStyle>
 <text>
 <![CDATA[<Center><B>$[name]</Center><br><br>$[description]</B><P><br>   Author:   $[Author_Year]<br/>
DOI:  $[DOI]<br>  Link  to  data:  $[Link_data]<br>    Link  to  image:  $[Link_image]<br>    Data  type:
$[DataType]<br>]]>
 </text>
 </BalloonStyle>
 </Style>
<Placemark>
<name> SB 1</name>
<description>
<![CDATA[Seismic stratigraphy and sediment thickness of the Nansen Basin, Arctic Ocean <br></br>
     <Center><img         src="https://dl.dropboxusercontent.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0"
width=400></Center>    <table border="1" padding="1">
    <tr><th>MAX offset(m)</th><th>WD</th><th>V(top)</th><th>V(bot)</th>
    <tr><td>7.3</td><td>0.77</td><td>1.7</td><td>1.7</td></tr>
     </table>
     SB - sonobuoy number; WD - water depth. Depths in km, velocities in km s-1.
    ]]></description>
               <ExtendedData>
               <Data name=    "Author_Year"><value>Engen et al. 2009</value> </Data>
               <Data     name="DOI"><value><a    href="https://doi.org/10.1111/j.1365-246X.2008.04028.x
">DOI</a></value> </Data>
               <Data name="Link_data"> <value>NA</value> </Data>
               <Data                             name="Link_image"><value><a                        href="
https://www.dropbox.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0 ">Image</a></value> </Data>
               <Data name="DataType"> <value> SB </value> </Data>
               </ExtendedData>
<styleUrl> #style </styleUrl>
               <Point>
                       <coordinates>4.31 ,81.65, -7.3</coordinates>
               </Point>
</Placemark>
<Placemark>
…
<name> SB 10</name>
<description>
<![CDATA[Seismic stratigraphy and sediment thickness of the Nansen Basin, Arctic Ocean <br></br>
     <Center><img         src="https://dl.dropboxusercontent.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0"
width=400></Center>    <table border="1" padding="1">
    <tr><th>MAX offset(m)</th><th>WD</th><th>V(top)</th><th>V(bot)</th>
    <tr><td>5.3</td><td>3.27</td><td>1.7</td><td>1.7</td></tr>
     </table>
     SB - sonobuoy number; WD - water depth. Depths in km, velocities in km s-1.
    ]]></description>
               <ExtendedData>
               <Data name=    "Author_Year"> <value>Engen et al. 2009</value> </Data>
               <Data    name="DOI">   <value><a    href="https://doi.org/10.1111/j.1365-246X.2008.04028.x
">DOI</a></value>
 </Data>
               <Data name="Link_data">  <value>NA</value>
 </Data>
```

```
               <Data                         name="Link_image"><value><a                         href="
https://www.dropbox.com/s/47tyygjd7xgz0th/Nans_son.jpg?dl=0 ">Image</a></value> </Data>
               <Data name="DataType">  <value> SB </value>  </Data>
               </ExtendedData> <styleUrl> #style </styleUrl>
               <Point>
                       <coordinates>13.82 ,82.92, -5.3</coordinates>
               </Point>
</Placemark>
</Document>
</kml>
```
Example of resulting KML file, see in folder M_point_all.kml and P_point_all.kml.

4.2. Polyline data

4.2.1. Polyline data. Typical input data

An example input text file for a first ten shots for multichannel seismic profile ‘BGR 13-108’ (longitude, latitude, elevation) looks like:

```
23.741681, 82.776997, 1.000000
23.743299, 82.776915, 1.000000
23.745034, 82.776822, 1.000000
23.746737, 82.776750, 1.000000
23.748417, 82.776671, 1.000000
23.749945, 82.776602, 1.000000
23.751601, 82.776526, 1.000000
23.753301, 82.776435, 1.000000
23.755006, 82.776362, 1.000000
23.756625, 82.776287, 1.000000
```

This is a typical example of input file for polyline data. Data file is presented in the folder with Examples: Data_line.txt.

4.2.2. Polyline data. Matlab script

This example:

- uploads some polyline locations from a text file,
- writes the parameters of polyline style (like color and size) to a KML file,
- sets a **description** of polyline.

Input file Data_line.txt and output file M_line.kml you can find in Examples folder.
```
clear
coo = [];
% Generate new KML 'Data_line.kml' file and feed it with lines information
filename = 'Data_line.kml';
% Assign the name of new polyline
name = 'Data_line';
% Open a file in write mode.
fid = fopen(filename, 'wt');

% Generate header for KML
header = ['<?xml version="1.0" encoding="UTF-8"?>',10,...
    '<kml xmlns="http://earth.google.com/kml/2.1">',10,...
    '<Document>', 10,...
    '<name>',10, name, 10,'</name>',10];

% Generate style of polyline
mystyle =  [ '<Style id="style">',10,...
'<LineStyle>',10,...
'<color>809932CC</color>',10,... %Lines color
'<width>1.3</width>',10,... %Lines width
'</LineStyle>',10,...
'</Style>',10];
```

```
xy_coo = load('..\Data_line.txt');

%Assign the value of coordinates and elevation of polyline
lat =  xy_coo(:,2);
lon =  xy_coo(:,1);
el  = xy_coo(:,3);
%Created a loop iteration over list of elements in file of coordinates
for i = 1:size(xy_coo,1)
%Set new coordinate list 'coo'
    coo = [coo, num2str(lon(i)),',', num2str(lat(i)),',', num2str(el(i)),10];
end
% Assign the attributes of KML file.
output = ['<Placemark>',10,...
    '<name>',name,'</name>',10,...
       '<description>',10,...
       ['<![CDATA[','Multichannel Seismic data','<br>',10,...
       ']]>'],10,...
       '</description>',10,...
      '<styleUrl>#style</styleUrl>',10,...
      '<LineString>',10,...
      '<coordinates>',10,...
      coo,...
      '</coordinates>',10,...
    '</LineString>',10,...
  '</Placemark>'];

% Generate footer for KML
footer = [10,'</Document>',10,...
    '</kml>'];

% Finally write the list of header, style, polylines, footer into KML file
fprintf(fid,'%s',header);
fprintf(fid,'%s',mystyle);
fprintf(fid,'%s',output);
fprintf(fid,'%s',footer);
fclose(fid);
```

File with MATLAB script is presented in the folder with Examples: M_line.m, resulting KML file is M_line.kml.

4.2.3. Polyline data. Matlab script. Additional image in the description

Assigns to KML points file with tag `<description>` a description with images from local computer or link's location from cloud. Input file is Data_line.txt, output file is M_line_image.kml from Examples folder.

```
'<description>',10,...
       ['<![CDATA[','Multichannel Seismic data','<br>',10,...
       '<img src="https://dl.dropboxusercontent.com/s/m2plgypsmok27wb/Bgr13_207_208.jpg?dl=0"
width=300>','<br>',10,...
       ']]>'],10,...
       '</description>',10,...
```

File with MATLAB script is presented in the folder with Examples: M_line_image.m, resulting KML file is M_line_image.kml.

4.2.4. Polyline data. Matlab script. Additional styles

Assigns attributes to profile line in KML file, which could be present inside a balloon on the map, appearing with click on location of profile line. Input file Data_line, output file M_line_all.kml.

```
% Assign the community approved KML elements values to polyline
authorList = 'Berglak et al. 2016';
bibLinks = '<a href="https://doi.org/10.3389/feart.2016.00091"> DOI</a>';
dataLink = '<a
href="http://www.bgr.bund.de/DE/Themen/MarineRohstoffforschung/Meeresforschung/Projekte/NIL/Berichte_
Expedition_Panorama1.htm">Cruise Report </a>';
imgLink = '<a href="https://www.dropbox.com/s/b7grzdlvyuicn9k/BGR13-208.jpg?dl=0"> Image Damm, </a><a
href="https://www.dropbox.com/s/m2plgypsmok27wb/Bgr13_207_208.jpg?dl=0">Image Berglak</a>';
datatype = 'MCS';

% Generate style of polylines and style of bubble window near line, that coming with click on map in
Google Earth and Fusion tables
 mystyle =  [ '<Style id="style">',10,...
          '<BalloonStyle>',10,...
          '<textColor>ff000000</textColor>',10,...
          '<text>',10,...
                '<![CDATA[<Center><B>$[name]</B></Center><br>$[description]<P><br/>', ...
                    'Author: $[Author_Year]<br/> DOI: $[DOI]<br/> Link to data: $[Link_data]<br>
Link to image: $[Link_image]<br/>Data type: $[DataType]]]>',10,...
           '</text>',10,'</BalloonStyle>',10,...
            '<LineStyle>',10,...
               '<color>809932CC</color>',10,... %Lines color
               '<width>1.3</width>',10,... %Lines width
            '</LineStyle>',10,...
            '</Style>',10 ];


# Assign the attributes of extended data.
output = ['<ExtendedData>',10,...
       '<Data name="Author_Year">',10,...
       '<value>',authorList,'</value>',10,...
      '</Data>',10,...
      '<Data name="DOI">',10,...
       '<value>','<href>',bibLinks,'</href>','</value>',10,...
      '</Data>',10,...
      '<Data name="Link_data">',10,...
                         '<value>','<href>',dataLink,'</href>','</value>',10,...
                     '</Data>',10,...
      '<Data name="Link_image">',10,...
       '<value>',imgLink,'</value>',10,...
      '</Data>',10,...
      '<Data name="DataType">',10,...
       '<value>',datatype,'</value>',10,...
       '</Data>',10,...
   '</ExtendedData>',10,...
];
```

File with complete MATLAB script is presented in the examples folder: M_line_all.m, resulting KML file is M_line_all.kml.

4.2.5. Polyline data. Python script

For programing in Python language, we used Anaconda navigator with Spyder 3.2.6. open source software. This example:

- uploads some polyline locations from a text file,
- writes the parameters of polyline style (like color and size) to a KML file, sets a **description** of polyline.

Input file Data_line.txt and output file P_line.kml you can find in Examples folder.

```
# Automatically added attributes for Python file
# -*- coding: utf-8 -*-
"""
# Create a new KML file
f = open("P_line.kml", "w")

#Assing a header and name of KML
header = "<?xml version=\"1.0\" encoding=\"UTF-8\"?> \n \
    <kml xmlns=\"http://earth.google.com/kml/2.1\"> \n \
    <Document> \n \
    <name> \n BGR13-108\n </name> \n"

# Generate style of polylines (color and width of line)
mystyle = "<Style id=\"style\"> \n \
<LineStyle> \n \
<color>80FFD700</color>    #Lines color\n\
<width>1.3</width>  #Lines width\n\
</LineStyle> \n\
</Style> \n"


#Upload NumPy package for scientific computing with Python
import numpy as np
```

```
#Load file ‘BGR13-108.txt‘ and make array of float
subtable = np.genfromtxt("..\Data_line.txt", delimiter = ',', dtype = np.float)

# Assign the header and style of polyline
f.write(str(header))
f.write(str(mystyle))

nz = np.size(subtable,0)

# Make array of numbers
subtable = np.array(subtable)
lat = subtable[:,0]
lon = subtable[:,1]
el = subtable[:,2]

# Description of point with images from modeling
descr = "<![CDATA[<Center><B>Multichannel Seismic data </B></Center><br>\
]]>"
# Assign the attributes of KML file name, description, extended data.
f.write("<Placemark>\n")
f.write("<name>" + "BGR13-108" + "</name> \n")
f.write("<description>\n" + descr + "</description>\n")
f.write("<styleUrl> #style </styleUrl> \n")
f.write("\t\t<LineString>\n")

# Set a loop iteration over list with latitude, longitude and elevation and write items to the list
of coordinate in KML
f.write("\t\t\t<coordinates>\n")
for i in range(nz):
    f.write(str(lat[i])+','+str(lon[i])+','+str(el[i])+' ')
f.write('\n\t\t\t</coordinates>\n')

f.write("\t\t</LineString>\n")
f.write("</Placemark>\n")

f.write("</Document>\n\n")
f.write("</kml>\n")
f.close()
```
File with Python script is presented in the folder with Examples: P_line.py, resulting KML file is P_line.kml.

4.2.6. Polyline data. Python script. Additional image in the description

Input file Data_line.txt and output file P_line_image.kml you can find in Examples folder.
```
# Description of polyline with image from modeling
descr = "<![CDATA[<Center><B>Multichannel Seismic data </B></Center><br>\
<img   src=\"https://dl.dropboxusercontent.com/s/m2plgypsmok27wb/Bgr13_207_208.jpg?dl=0\"   width=300
><br> \n\
Composite MCS line depicting the northeastern Yermak Plateau, the southwestern part of the Nansen
Basin, and the North Barents Sea continental margin.<br> \n\
    ]]>"
```
File with complete Python script is presented in the folder with Examples: P_line_image.py, resulting KML file is P_line_image.kml.

4.2.7. Polyline data. Python script. Additional styles

Assigns attributes to polyline, which could be present inside a balloon on the map, appearing with click on line. Input file Data_line, output file P_line_all.kml.
```
# Generate style of polylines and style of bubble window near line, that coming with click on map in
Google Earth, Fusion tables or website
mystyle = "<Style id=\"style\"> \n \
<BalloonStyle> \n \
<text> \n \
<![CDATA[<Center><B>$[name]</B></Center><br><br>$[description]<P><br> \
Author:  $[Author_Year]<br/>  DOI:  $[DOI]<br>Link  to  data:  $[Link_data]<br>  Link  to  image:
$[Link_image]<br>  Data type: $[DataType]<br>  Color: $[Color]]]> \n \
</text> \n \
</BalloonStyle> \n \
<LineStyle> \n \
<color>80FFD700</color>    #Lines color\n\
<width>1.3</width>  #Lines width\n\
</LineStyle> \n\
</Style> \n"

# Assign the community approved KML elements values to polyline
authorList = 'Berglak et al. 2016'
bibLinks = '<a href=https://doi.org/10.3389/feart.2016.00091 > DOI</a>'
dataLink = '<a
href=http://www.bgr.bund.de/DE/Themen/MarineRohstoffforschung/Meeresforschung/Projekte/NIL/Berichte_E
xpedition_Panorama1.htm > Cruise Report, </a>'
imgLink = '<a href=https://www.dropbox.com/s/b7grzdlvyuicn9k/BGR13-208.jpg?dl=0 >Image Damm</a>'
datatype = 'MCS'

# Assign the attributes of extended data
f.write(("\t\t<ExtendedData> \n")+ \
("\t\t<Data name=\t\"Author_Year\"> \n" + "<value>" + authorList + "</value>\n </Data>\n") + \
("\t\t<Data name=\"DOI\">\n <value>" + bibLinks + "</value> \n </Data> \n")+ \
("\t\t<Data name=\"Link_data\"> \n <value>" + dataLink + "</value>\n </Data> \n")+ \
("\t\t<Data name=\"Link_image\"> \n <value>"+ imgLink + "</value>\n </Data> \n")+ \
("\t\t<Data name=\"DataType\"> \n <value>" + datatype + "</value> \n </Data>\n ") + \
("\t\t<Data name=\"Color\"> \n <value>"+ color + "</value>\n </Data> \n")+ \
("\t\t</ExtendedData> \n"))
```

File with complete Python script is presented in the examples folder: P_line_all.py. Example of resulting KML file, see in folder P_line_all.kml.

4.2.8. Polyline data. Resulting KML file

At the result of addicting of complete metadata with help of Matlab (M_line_all.m) or Python (P_line_all.py) we have got KML file for sonobuoys (points) with name, description, image, table, author and year of publication, DOI, links to depository, type of data. Below we present resulting short version of KML point file with the first and second sonobuoy. Se in Example folder files M_line_all.kml and P_line_all.kml of KML files.

```
<?xml version="1.0" encoding="UTF-8"?>
    <kml xmlns="http://earth.google.com/kml/2.1">
    <Document>
    <name>  BGR13-108 </name>
<Style id="style">
 <BalloonStyle>
 <text><![CDATA[<Center><B>$[name]</B></Center><br><br>$[description]<P><br>  Author:  $[Author_Year]<br/>  DOI:  $[DOI]<br>Link  to
data: $[Link_data]<br> Link to image: $[Link_image]<br>  Data type: $[DataType]<br>]]>
 </text>
 </BalloonStyle>
 <LineStyle>
 <color>80FFD700</color>    #Lines color
<width>1.3</width>  #Lines width
</LineStyle>
</Style>
<Placemark>
<name>BGR13-108</name>
<description>
<![CDATA[<Center>      <B>      Multichannel      Seismic      data      </B>      </Center>      <br>      <img      src=
"https://dl.dropboxusercontent.com/s/m2plgypsmok27wb/Bgr13_207_208.jpg?dl=0"  width=300  >  <br>  Composite  MCS  line  depicting  the
northeastern Yermak Plateau, the southwestern part of the Nansen Basin, and the North Barents Sea continental margin.<br>
   ]]></description>
<styleUrl> #style </styleUrl>
                        <ExtendedData>
                        <Data name=          "Author_Year">
<value>Berglak et al. 2016</value> </Data>
                        <Data name="DOI">
 <value><a href="https://doi.org/10.3389/feart.2016.00091"> DOI</a></value>  </Data>
                        <Data name="Link_data">
 <value><a
href="http://www.bgr.bund.de/DE/Themen/MarineRohstoffforschung/Meeresforschung/Projekte/NIL/Berichte_Expedition_Panorama1.htm">    Cruise
Report, </a></value> </Data>
                        <Data name="Link_image">
 <value><a href="https://www.dropbox.com/s/b7grzdlvyuicn9k/BGR13-208.jpg?dl=0">Image Damm</a></value> </Data>
                        <Data name="DataType">
 <value>MCS</value>  </Data>
                        </ExtendedData>
                        <LineString>
                                    <coordinates>
```

```
23.741681,82.776997,1.0        23.743299,82.776915,1.0        23.745034,82.776822,1.0        23.746737,82.77675,1.0        23.748417,82.776671,1.0
23.749945,82.776602,1.0 23.751601,82.776526,1.0 23.753301,82.776435,1.0 23.755006,82.776362,1.0 23.756625,82.776287,1.0
                              </coordinates>
                    </LineString>
</Placemark>
</Document>

</kml>
```
Example of resulting KML file, see in folder M_line_all.kml and P_line_all.kml.

4.3. Conversion KML file

4.3.1. From SHP to KML

Easy way is open SHP file in Google Earth and save it like KML.

4.3.2. From KML to GPML

5. Useful applications for work with database

For begin to produce big massive of data in KML for web-database we propose below application to install on computer. Choose based your preference.

5.1. Google Fusion Tables

Fusion Tables (FT) is a data visualization web application to gather, visualize, and share data tables. It's needs google account to use FT.

5.1.1. Upload of FT on Google account

User does not need to install FT on the computer, but it is necessary to activate Fusion tables on Google account. Follow description bellow (Fig. 4 and Fig. 5) to do it.

At the first, open Drive disk on Google account.

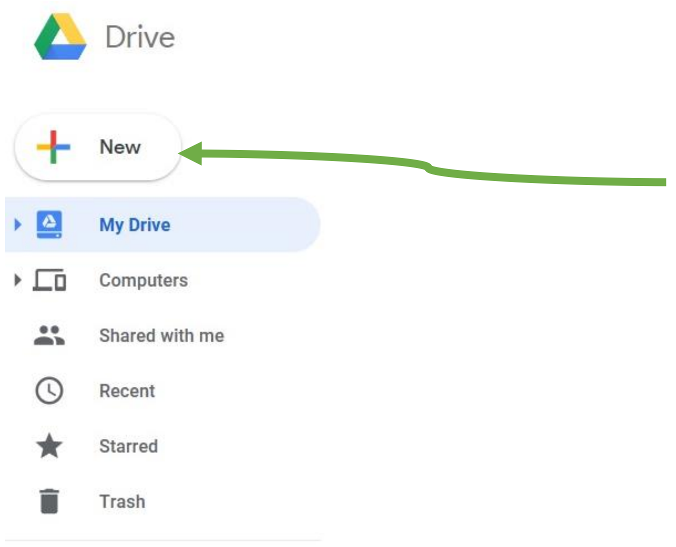

Drive
New
My Drive
Computers
Shared with me
Recent
Starred
Trash
Open Drive disk on Google account.
To upload FT application click on New button

Figure 4. Screenshot from Google account Drive disk.

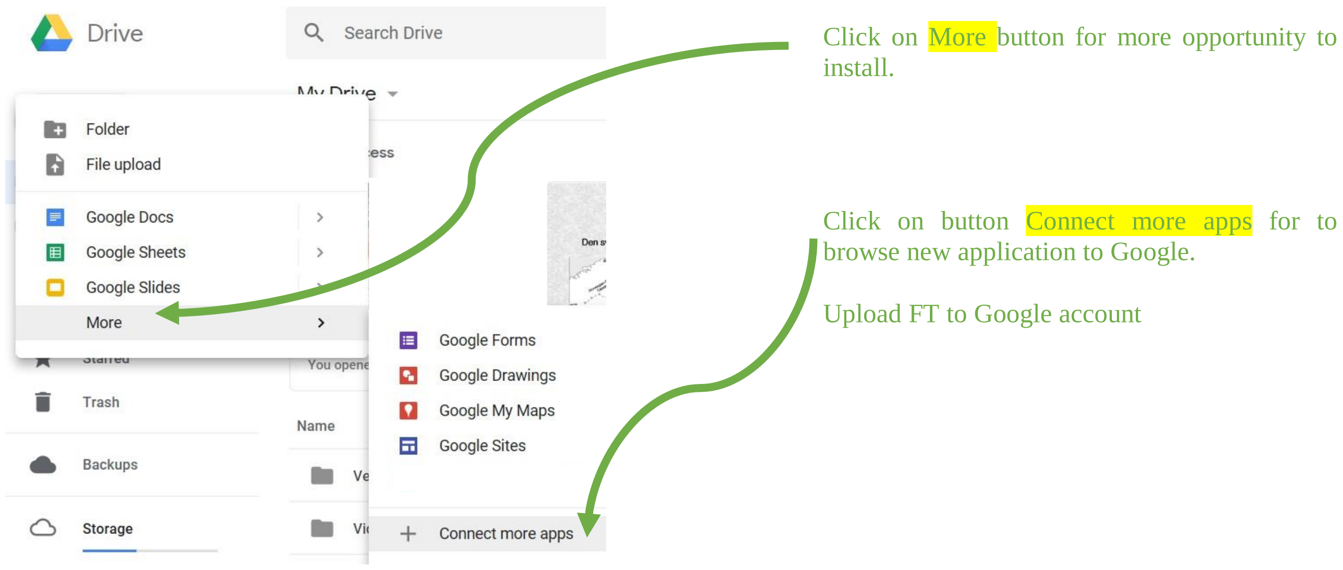
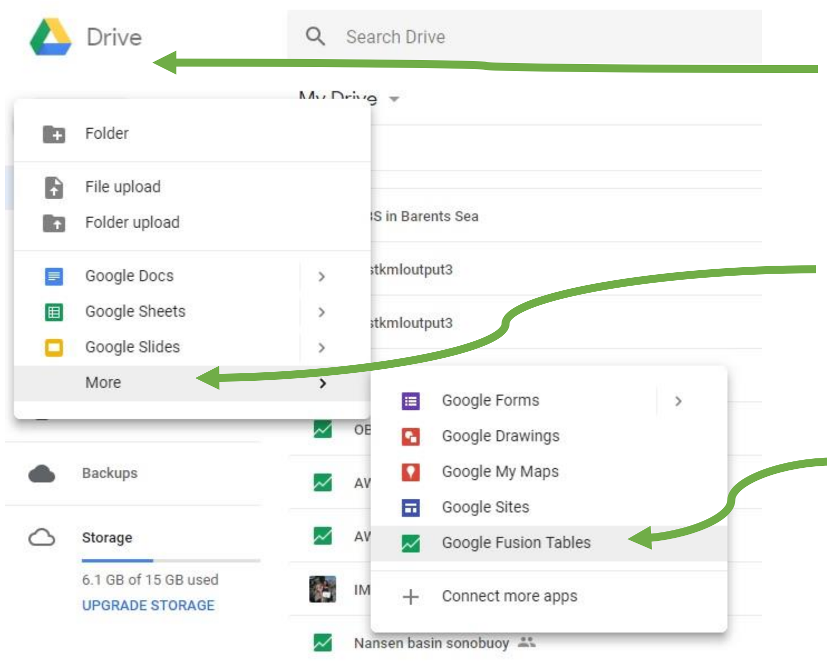


Figure 5. Screenshot from Google account installation of new application.

It is necessary to browse application once and you get it on you Google account (Fig. 5) for next time.

5.1.2. Import data

For working with FT open again Google Drive disk on you Google account and create new Google Fusion Tables (Fig. 6). When upload data tables from computer or cloud. Importing data can be like spreadsheets, delimited text files (txt, CSV and another) or KML files (Fig. 7-9). See examples below step by step how to upload KML file in FT (Fig. 7-9).

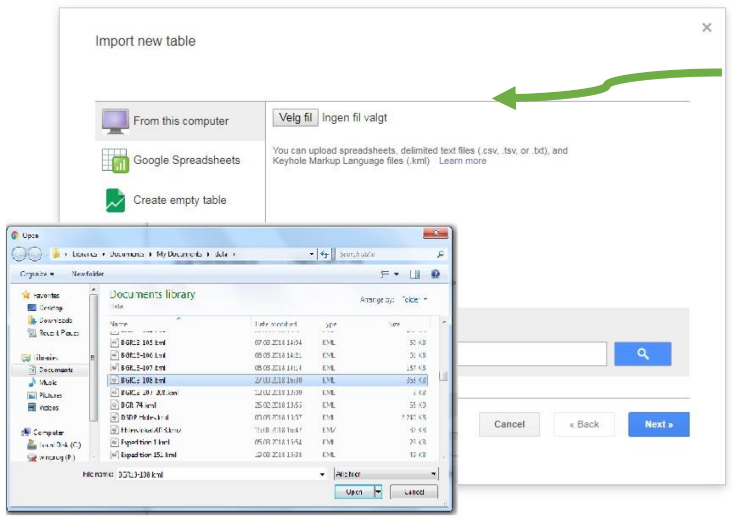


Open Drive disk on Google account.

Click on button More.

Click on Google Fusion Tables button for create new table.

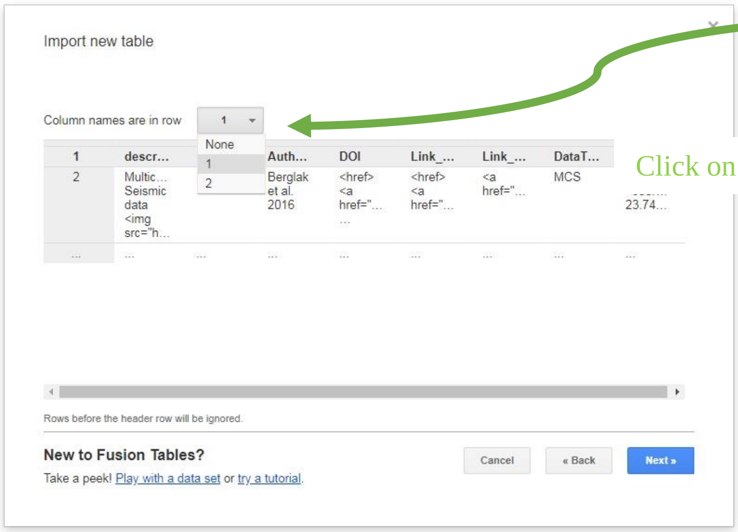
Fig. 6. First step. Creation of new table in Fusion tables



You can upload spreadsheets, delimited text files (.csv, .tsv, or .txt), and Keyhole Markup Language files (.kml) Learn more

Choose a Browse file on computer for import KML file to FT

Fig. 7. Second step. Upload KML file profile BGR13-108 with Multichannel seismic data to FT.


Import new table

Column names are in row: 1 (None / 1 / 2)

1	descr...		Auth...	DOI	Link_...	Link_...	DataT...	
2	Multic... Seismic data <img src="h...		Berglak et al. 2016	<href> <a href="... ...	<href> <a href="...	<a href="...	MCS	23.74...
...	...	...	...	...	...	...	...	...

Rows before the header row will be ignored.

New to Fusion Tables?
Take a peek! Play with a data set or try a tutorial.

Preview the columns being imported. Choose a row to present name of column in table. Click Next

Click on Rows in FT

Fig. 8. Third step. Automatically choice of row and column on the import KML file in FT.

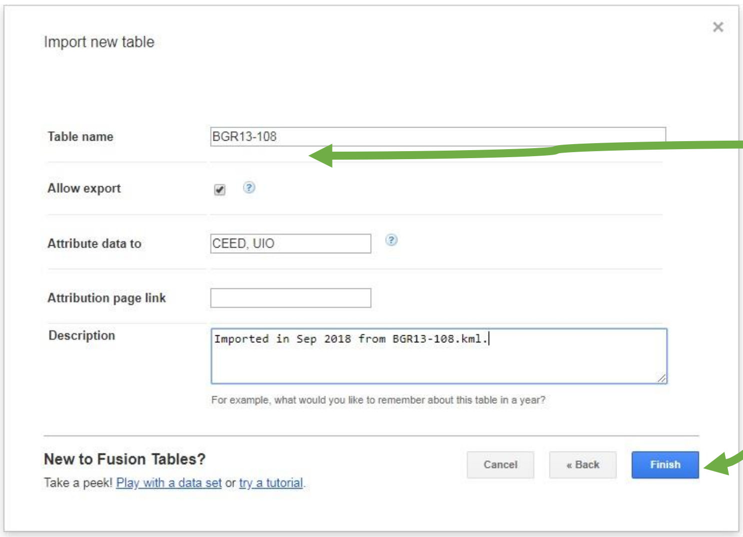

Import new table

Table name	BGR13-108
Allow export	✓
Attribute data to	CEED, UIO
Attribution page link	
Description	Imported in Sep 2018 from BGR13-108.kml.

For example, what would you like to remember about this table in a year?

New to Fusion Tables?
Take a peek! Play with a data set or try a tutorial.

Assign a new table's name. Edit attribution and description as needed.

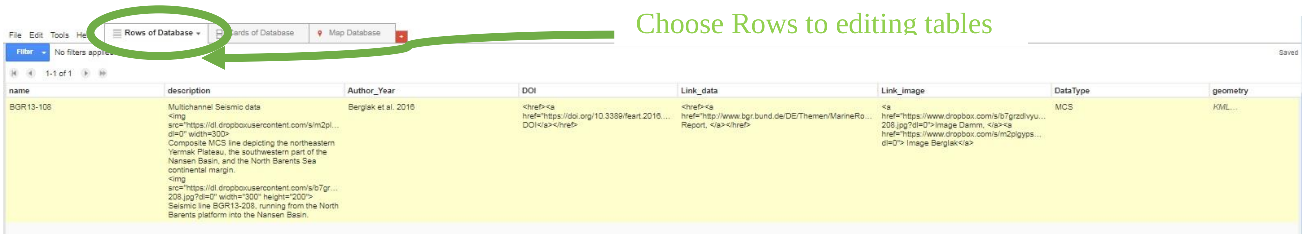
Click a Finish for come to next step

Fig. 9. Fourth last step. Assigning of new table name, here 'BGR13-108'

5.1.3. Data visualization on map

When we download KML file in FT, we can use application to insert, update, delete and query data.

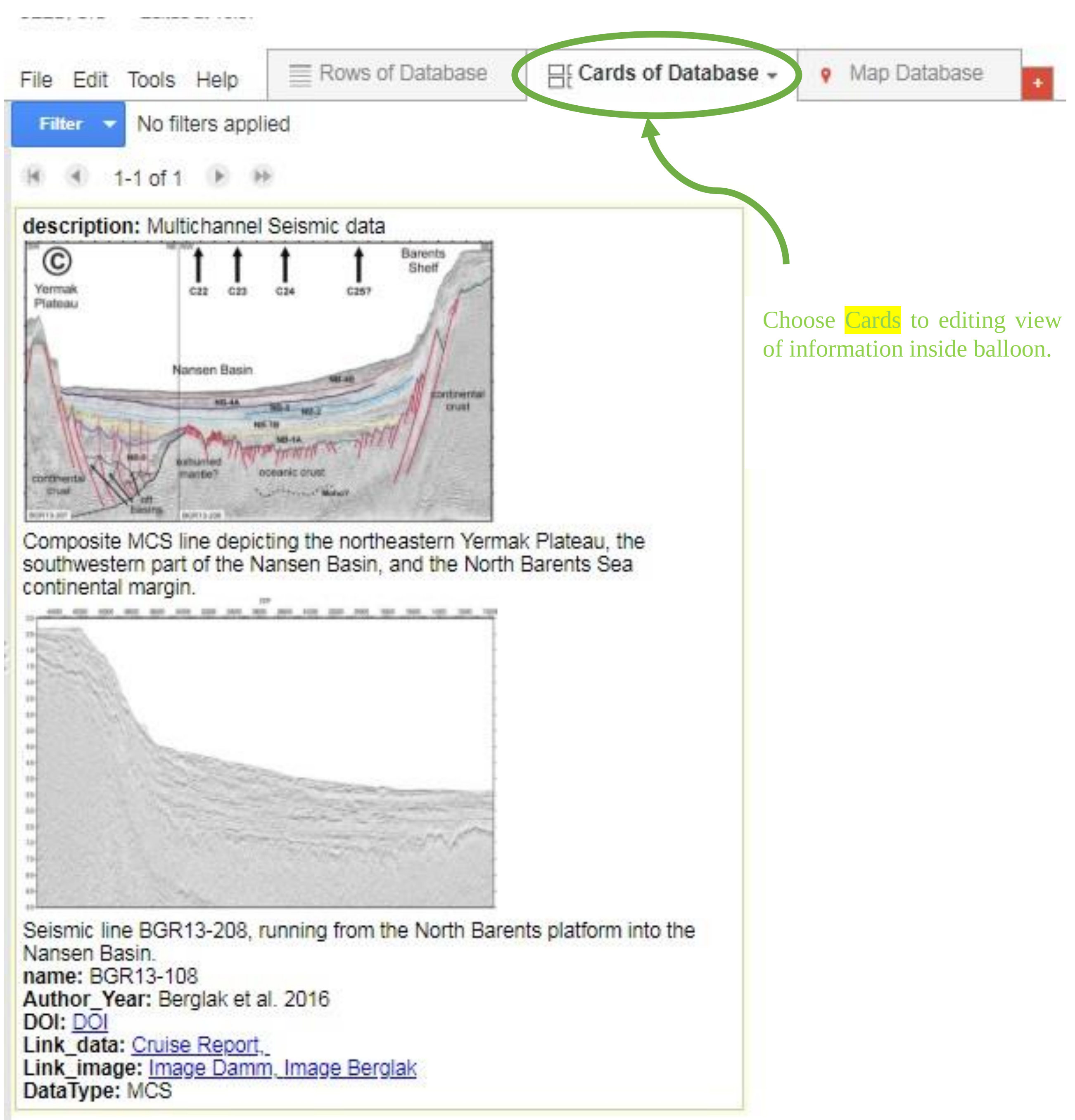
Choose Rows to editing tables



name	description	Author_Year	DOI	Link_data	Link_image	DataType	geometry
BGR13-108	Multichannel Seismic data <img src="https://dl.dropboxusercontent.com/s/m2pl... dl=0" width=300> Composite MCS line depicting the northeastern Yermak Plateau, the southwestern part of the Nansen Basin, and the North Barents Sea continental margin. <img src="https://dl.dropboxusercontent.com/s/b7gr... 208.jpg?dl=0" width="300" height="200"> Seismic line BGR13-208, running from the North Barents platform into the Nansen Basin.	Berglak et al. 2016	<href><a href="https://doi.org/10.3389/feart.2016.... DOI</a></href>	<href><a href="http://www.bgr.bund.de/DE/Themen/MarineRo... Report, </a></href>	<a href="https://www.dropbox.com/s/b7grzdlvyu... 208.jpg?dl=0">Image Damm, </a><a href="https://www.dropbox.com/s/m2plgyps... dl=0"> Image Berglak</a>	MCS	KML...

Fig. 10. Presentation of uploaded row in Fusion Tables, where name of columns presented of balloon elements. Multichannel seismic profile BGR13-108.

Filtering of the data cards according to their attributes present in Figure 11.



description: Multichannel Seismic data

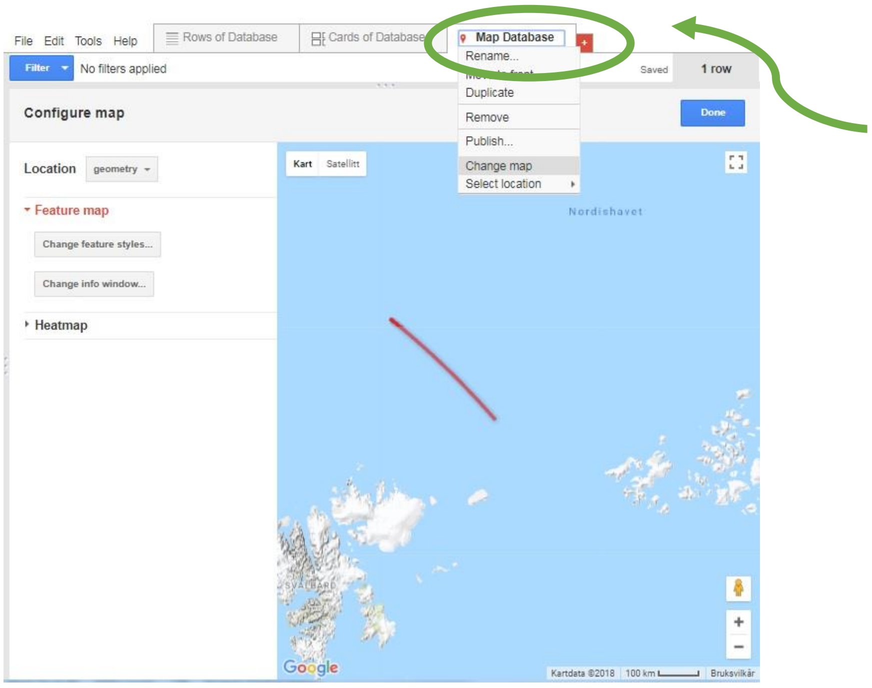
Composite MCS line depicting the northeastern Yermak Plateau, the southwestern part of the Nansen Basin, and the North Barents Sea continental margin.

Seismic line BGR13-208, running from the North Barents platform into the Nansen Basin.
name: BGR13-108
Author_Year: Berglak et al. 2016
DOI: DOI
Link_data: Cruise Report,
Link_image: Image Damm, Image Berglak
DataType: MCS

Choose Cards to editing view of information inside balloon.

Fig. 11. Presentation of uploaded card in FT with description (name and image) and balloon elements (Author, DOI, Link to data, Link to image and Data type).

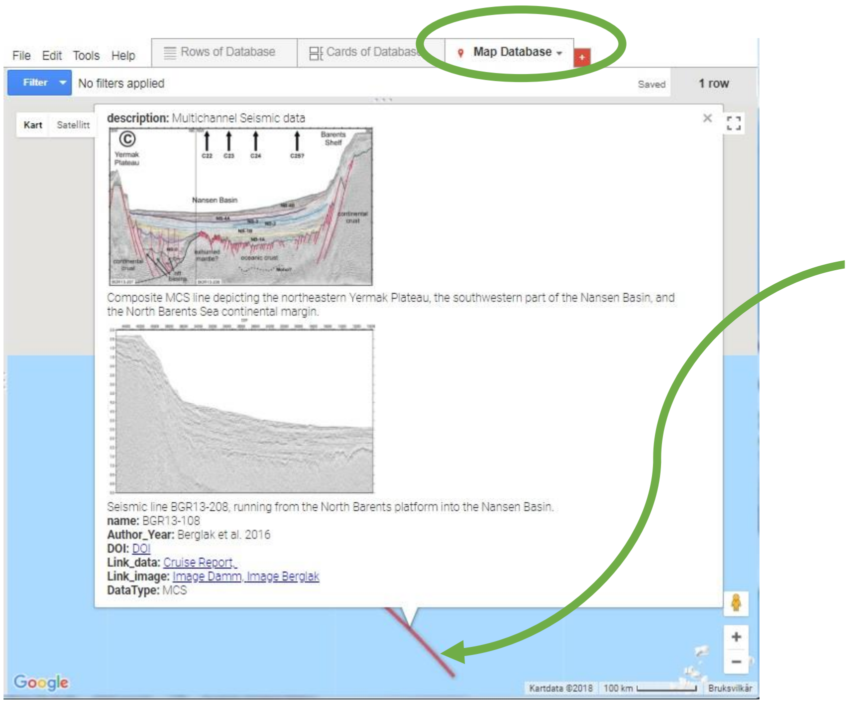
Make a map in minutes. Turn location tables into maps. Share map. Same structure of elements in FS like for KML files. See the data on a map (Fig. 12) or as a chart (Fig. 10) immediately. Filtrate table for more selective visualizations.



Choose Map, click change map to editing visualization on map of drawn objects

Fig. 12. Presentation of uploaded data on the map in FT. Red line is Multichannel seismic profile in the Western Arctic region, line BGR13-108.

This map changes when you change your table's data.
For more information about seismic profile click on the red line on map, when appeared balloon with description and available data.



description: Multichannel Seismic data

Composite MCS line depicting the northeastern Yermak Plateau, the southwestern part of the Nansen Basin, and the North Barents Sea continental margin.

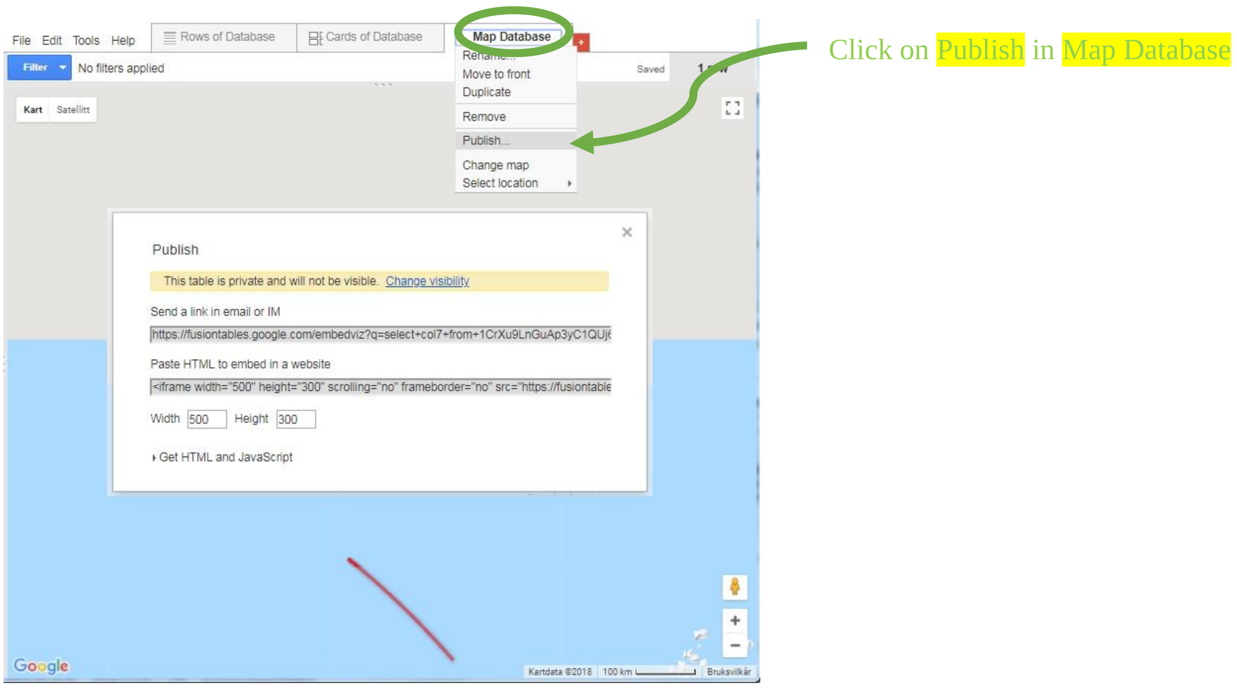
Seismic line BGR13-208, running from the North Barents platform into the Nansen Basin.
name: BGR13-108
Author_Year: Berglak et al. 2016
DOI: DOI
Link_data: Cruise Report,
Link_image: Image Damm, Image Berglak
DataType: MCS

Click on red line and get information about multichannel seismic profile BGR13-108.

Fig. 13. Presentation of map with uploaded seismic data in FT. Red line is seismic profile, line BGR13-108, in the balloon is a KML elements from cards of FT.

5.1.4. Sharing and publishing maps in website

Than database map filled with chart of data, it is feasibly embed it in a web page or blog post (Fig. 14). We showed resulting web database on website https://norramarctic.wordpress.com/resources-and-links/, see the second chapter of Cookbook (Fig. 1). Below presents description to share Map view from Fusion tables.



https://fusiontables.google.com/embedviz?q=select+col7+from+1CrXu9LnGuAp3yC1QUj

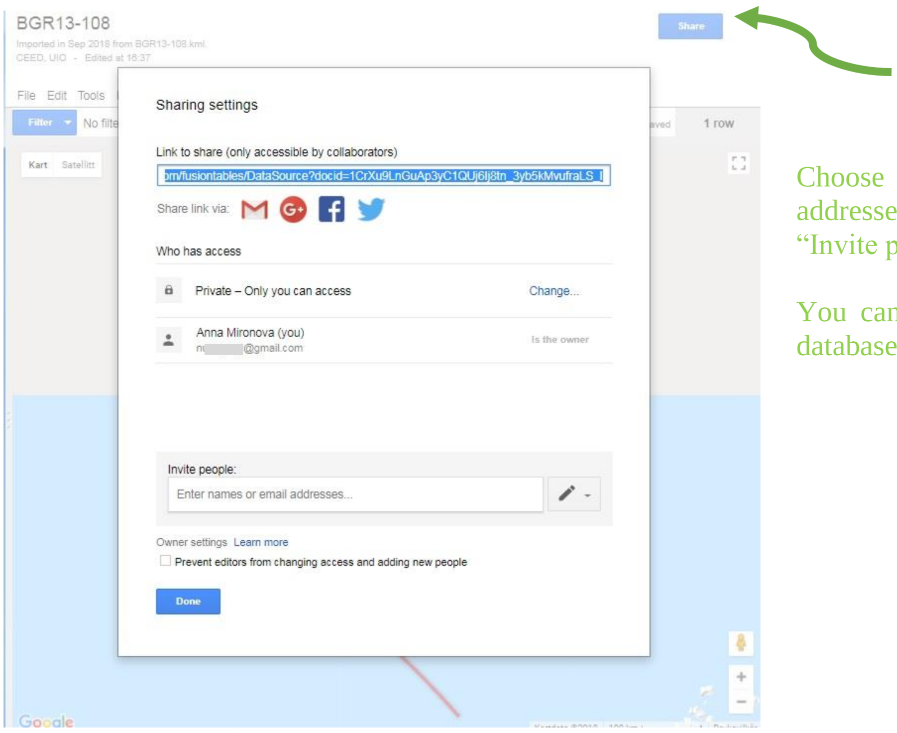
<iframe width="500" height="300" scrolling="no" frameborder="no" src="https://fusiontable

Width 500 Height 300

Fig. 14. Print view of FT.

To share the map's URL you should copy the "Send a link in email or IM" URL. After that paste the URL into an email or another browser to share as you want. To share the map in an embeddable frame you should copy the "Paste HTML to embed in a website" HTML code and change the width and height as needed. Paste the code into a webpage.

It easy to sharing intermediate and final results with colleges, for example to send a link by email (Fig. 15). It will be always display the latest data values from Fusion tables and helps communicate more easily. You can specify the availability of the database for editing or for reading only.



Sharing settings

Link to share (only accessible by collaborators)

om/fusiontables/DataSource?docid=1CrXu9LnGuAp3yC1QUj6lj8tn_3yb5kMvufraLS_

Share link via:

Who has access

Private – Only you can access — Change...

Anna Mironova (you) n[redacted]@gmail.com — Is the owner

Invite people:

Enter names or email addresses...

Owner settings Learn more

Prevent editors from changing access and adding new people

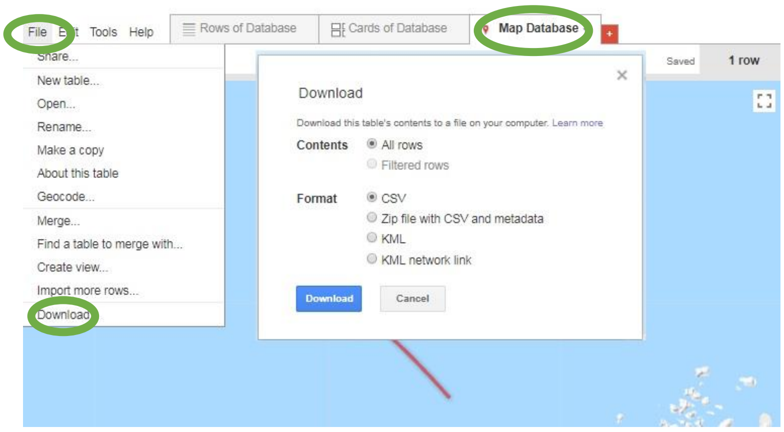
Done

Fig. 15. Print view from FT sharing settings.

Choose button Share and write email addresses of collaborators in the window “Invite people”.

You can specify the availability of the database for editing or for reading only.

The same way to export web database from FT as link, CSV or KML files present in Fig. 16.



Download

Download this table's contents to a file on your computer. Learn more

Contents: (●) All rows; ( ) Filtered rows

Format: (●) CSV; ( ) Zip file with CSV and metadata; ( ) KML; ( ) KML network link

Download Cancel

Fig. 16. Print view of FT for download information from database in different formats.

Choose File and click Download.
To share a static copy of the map's KML. Click KML.

To share a dynamic link to the map's KML.
Click KML network link

5.2. Google Earth

Google Earth (GE) is a free, powerful yet simple tool for viewing information geographically - whether it is analysing change over time, view historical imagery, navigate to a city with the directions module, import shape files, geocode addresses, and create a route. How to work with GEPro, find in tutorial.
Link to download https://www.google.com/earth/

5.2.1. Map

Fullness of the database at this stage of work shows in the Fig. 17. Here database presented in polar stereographical coordinates there North Pole in the middle of the map.

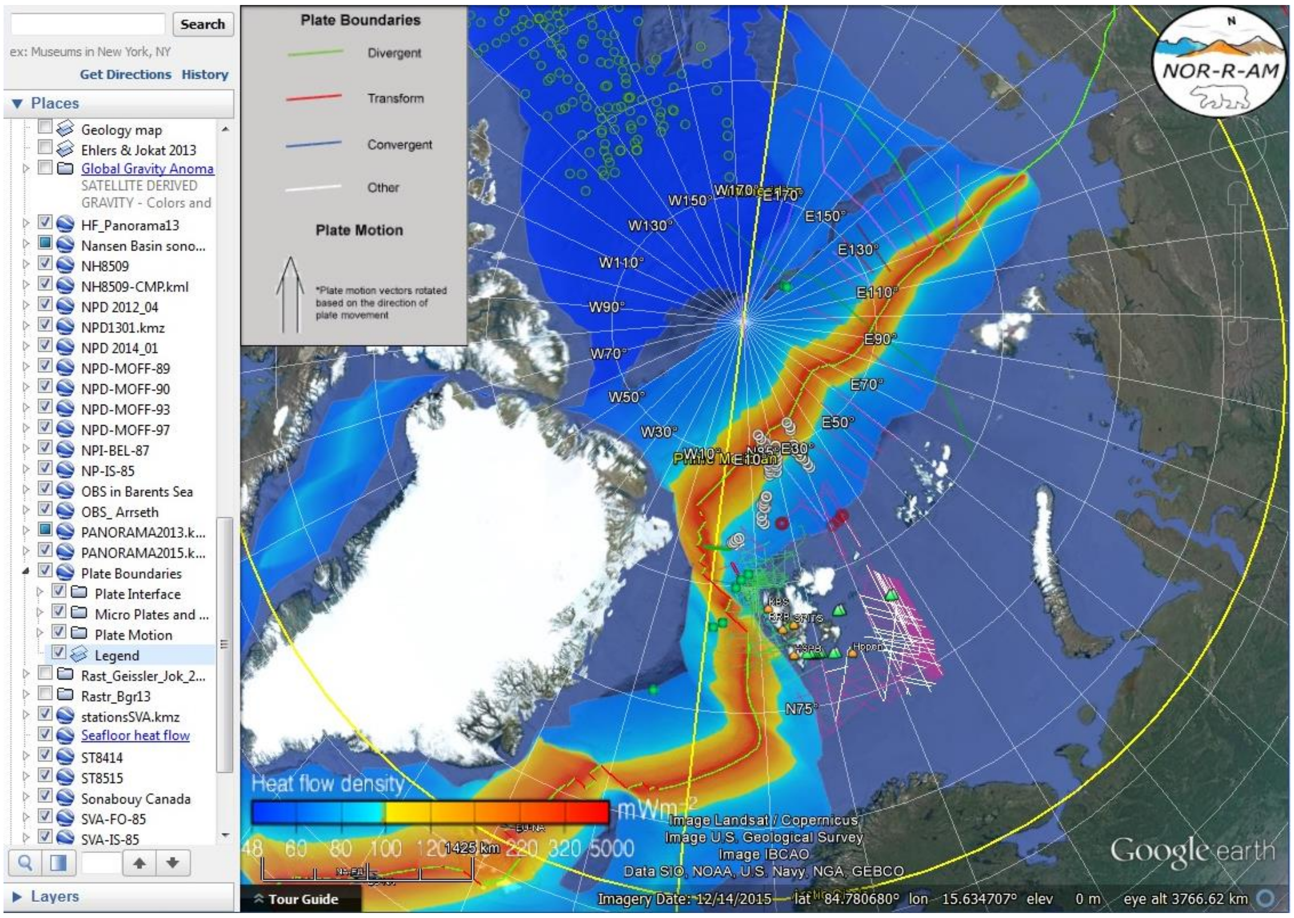
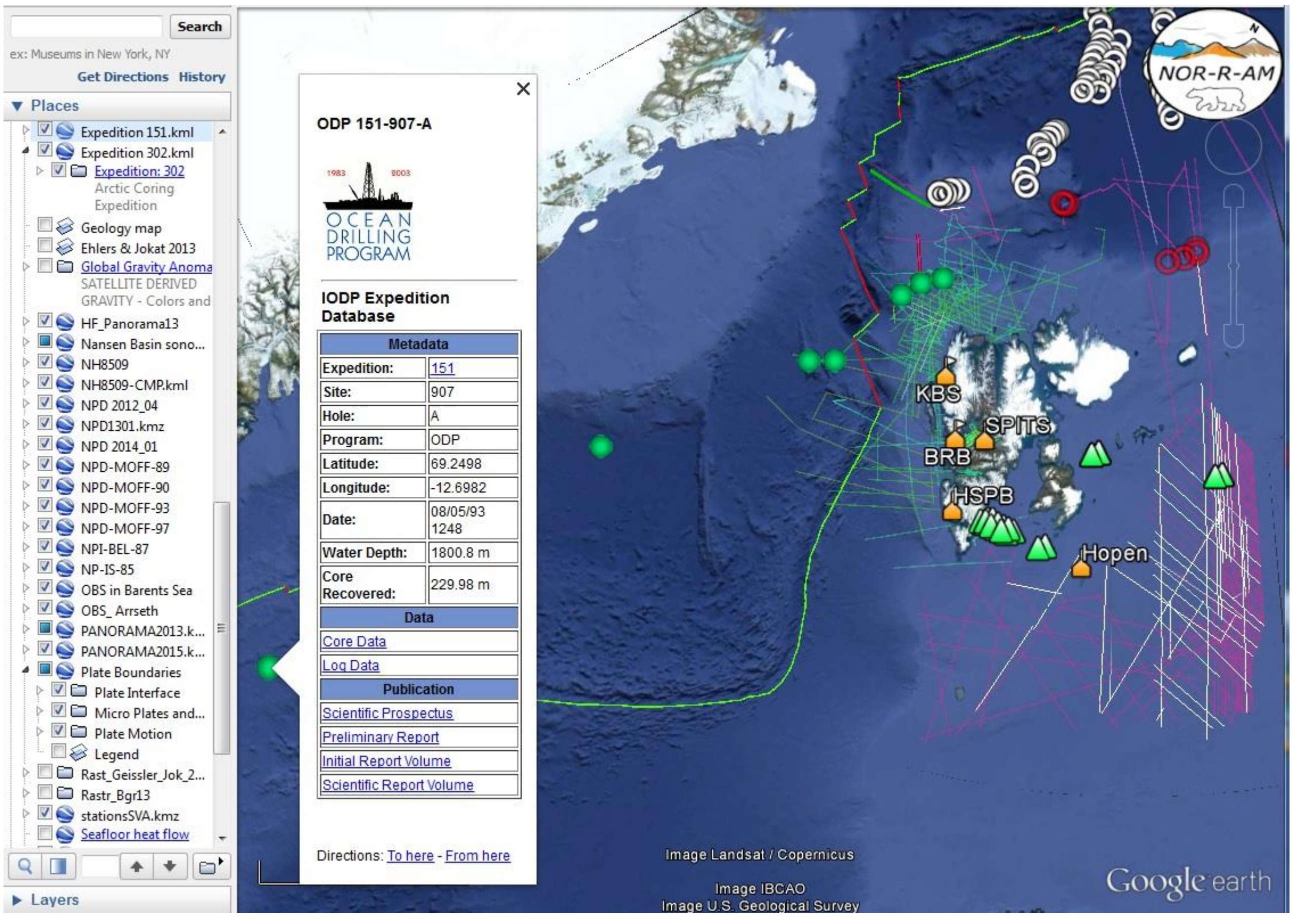


Fig. 17. Screen view from Google Earth program with overview of database information in the Western Arctic region. Color map of Seafloor of heat flow density made by Labrous et al. Heat flow measurement's stations present on the map with red rings, see Cruise report by Damm et al. Sonobuoys data shown also by rings, where green ring are the Canadian basin and white are the Nansen basin. Location of MCS data show by lines for references see KML files and supplementary material. Green circles present drilling well from the International Ocean drilling program (IODP). Green triangles are OBS data. Oranges label are permanent/temporary seismic stations. Plate boundaries see legend.



ODP 151-907-A

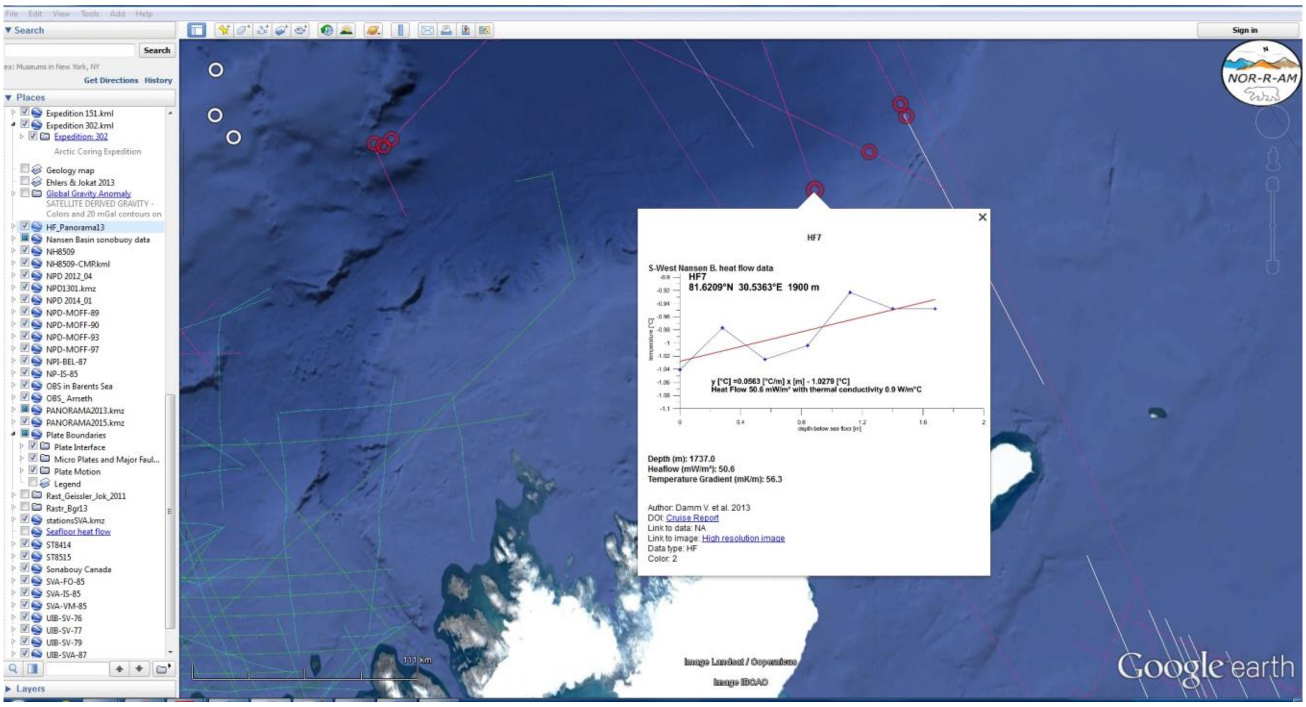
1983 2003 OCEAN DRILLING PROGRAM

IODP Expedition Database

Metadata	
Expedition:	151
Site:	907
Hole:	A
Program:	ODP
Latitude:	69.2498
Longitude:	-12.6982
Date:	08/05/93 1248
Water Depth:	1800.8 m
Core Recovered:	229.98 m
Data	
Core Data	
Log Data	
Publication	
Scientific Prospectus	
Preliminary Report	
Initial Report Volume	
Scientific Report Volume	

Directions: To here - From here

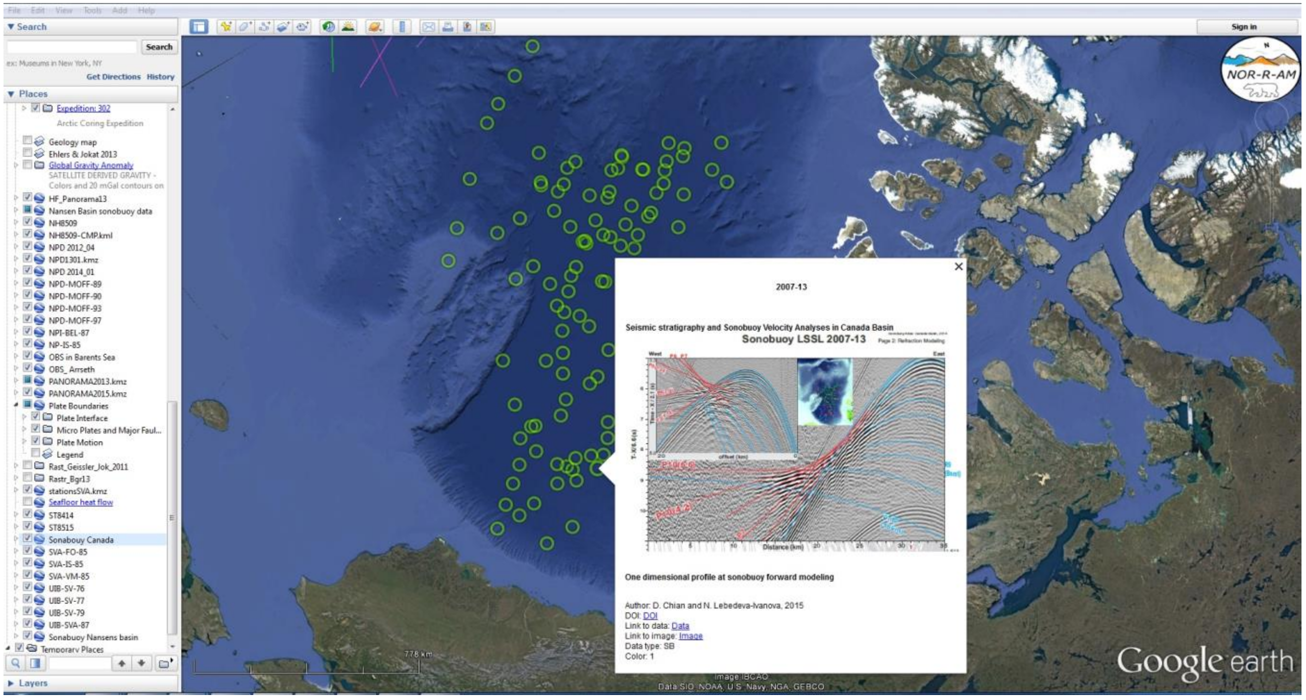
Fig. 18. Screen view from GE, in the balloon are presented metadata with the geological sampling of a drilling well from the expedition database in the Svalbard region. Data uploaded from IODP (the International Ocean drilling program), website.


HF7
S-West Nansen B. heat flow data
HF7
81.6209°N 30.5363°E 1900 m
temperature [°C]
depth below sea floor [m]
y [°C] =0.0563 [°C/m] x [m] - 1.0279 [°C]
Heat Flow 50.6 mW/m² with thermal conductivity 0.9 W/m°C
NOR-R-AM
Image Landsat / Copernicus
Image IBCAO
Google earth

Depth (m): 1737.0
Heaflow (mW/m²): 50.6
Temperature Gradient (mK/m): 56.3

Author: Damm V. et al. 2013
DOI: Cruise Report
Link to data: NA
Link to image: High resolution image
Data type: HF
Color: 2

Fig. 19. Screen view from GE, in the balloon are presented heat flow measurements in the South-West of Nansen basin.



2007-13

Seismic stratigraphy and Sonobuoy Velocity Analyses in Canada Basin

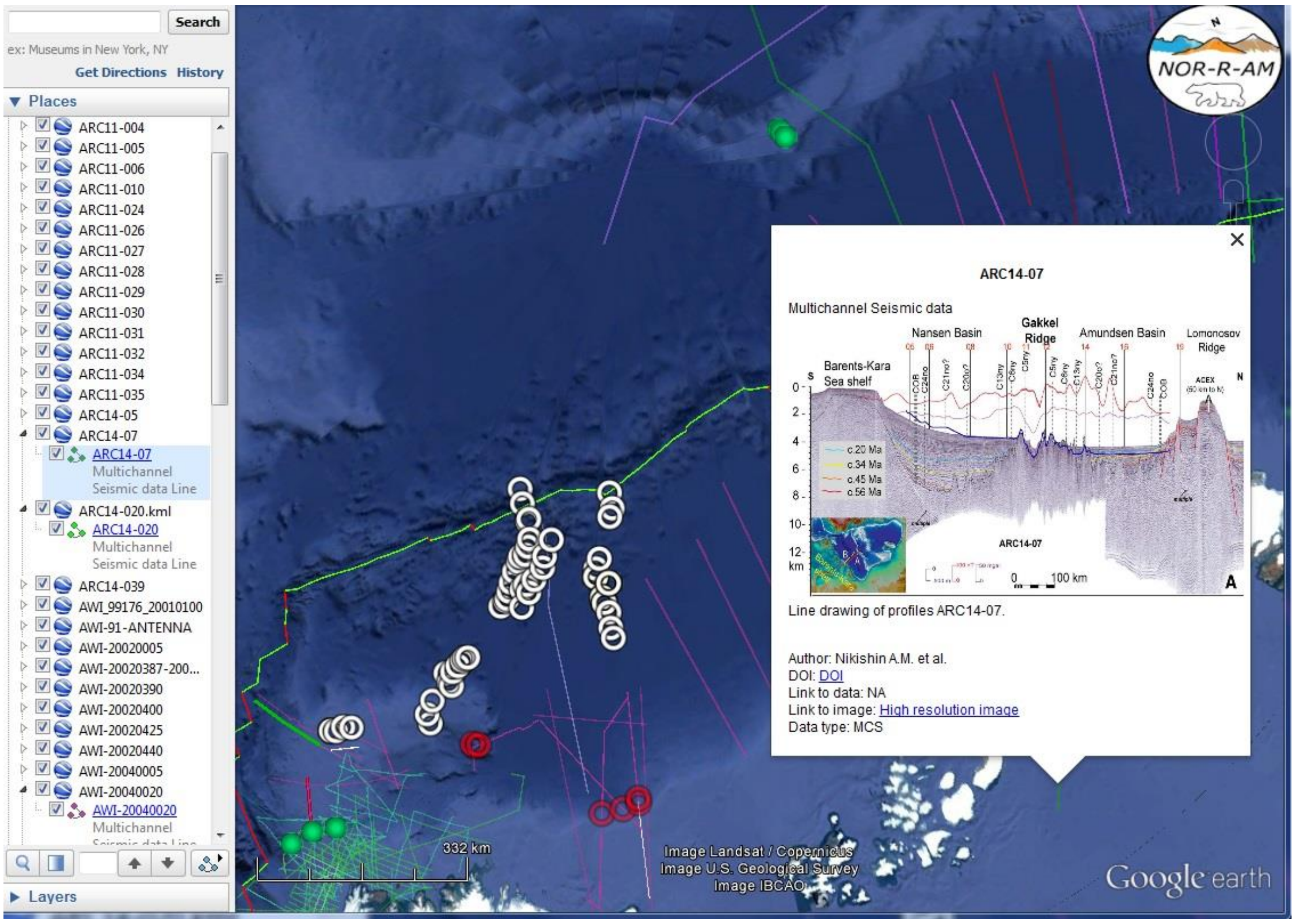
Sonobuoy LSSL 2007-13

Page 2: Refraction Modeling

One dimensional profile at sonobuoy forward modeling

Author: D. Chian and N. Lebedeva-Ivanova, 2015
DOI: DOI
Link to data: Data
Link to image: Image
Data type: SB
Color: 1

Fig. 20. Screen view from GE, in the balloon are presented seismic stratigraphy and sonobuoys velocity analyses in the Canada Basin.



ARC14-07

Multichannel Seismic data

Line drawing of profiles ARC14-07.

Author: Nikishin A.M. et al.
DOI: DOI
Link to data: NA
Link to image: High resolution image
Data type: MCS

Fig. 21. Screen view from GE, in the balloon are presented Multichannel seismic data profile ARC14-07 through the Barents-Kara sea shelf, the Nansen basin, the Gakkel ridge, the Amundsen basin until the Lomonosov ridge.

5.3. Anaconda distribution for Python

Anaconda is a free and open source navigator distribution of the Python programming languages for data science and machine learning related applications that aims to simplify package management and deployment. Through Anaconda navigator we can activate Spyder environment for further programing.
Link to download https://www.anaconda.com/download/

5.4. MATLAB/Octave

MATLAB is a commercial multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. Link to download:
https://se.mathworks.com/campaigns/products/trials.html?s_eid=ppc_30720711922&q=matlab
Octave is a scientific programming language. Powerful mathematics-oriented syntax with built-in plotting and visualization tools. Free software, runs on GNU/Linux, macOS, BSD, and Windows. Link to download:
https://www.gnu.org/software/octave/#install

5.4.1.Comparing Matlab and Python scripts

In cookbook, we used both MATLAB and Spyder computing environment. Both are satisfied our requests, but Spyder is a charge free program. MATLAB and Python languages have a lot in common, but there are some differences. The table below gives some equivalents that it has been used for programming in MATLAB and Python.

Table 1. Purpose equivalents

MATLAB	**PYTHON**	**Notes**
clear	# -*- coding: utf-8 -*-	Before proceeding, we need to clear the workspace by typing
;		To suppress the display of an array or the result of an operation in general, the line in MATLAB should be ended with a semicolon.
…	\	Than script’s line not ended and rest be placed in next line
10	/n	In KML should be write new line position
num2str()	str( )	Numerical values converts into character string
size	np.size	Information about variables sizes or dimensions, for NumPy package in Python
load( )	open( )	Import the data from file with the command.
fid=(filename,‘wt’)	f=open(filename,“w”)	Used to read the text from the file using textscan and to write it into the cell array
for i= 1:size( )	for i in range( )	Loop from 1 to size or range of array
fprint( )	f.write( )	Write in the file

fclose( )	f.close( )	Close the file

6. Conclusion

We introduced a structure for facilitating work and cataloguing information. All elements for KML files contains specified structure with description of Authors of publication and year, Digital Object Identifier (DOI), Live link to data source server/archive and published seismic image/interpretation, Data type. The web-database is constantly updated and its latest version is always available on the project website https://norramarctic.wordpress.com/resources-and-links/.

Web-GIS for Arctic was created with full data access for:

- Modification
- Upload
- Download
- Edition.

Different data types were applied as:

- Lines
- Points
- Geo-reference
- Images overlay
- Including recent seismic lines & vintage data

Web-GIS database can be useful for data coverage in article also Elsevier allows to submit interactive maps. For the present, we have included into this project several raster data, lines and point information:

- Active-source seismic,
- Reflection and refraction profiles/crustal thickness,
- Velocities distributions with description of interpretation for lines,
- Seismic images (451 profiles),
- Permanent/temporary seismic stations (5 stations),
- Heat flow measurements (7 stations),
- Earthquake locations and magnitudes,
- Information on geological sampling (e.g. Gakkel Ridge, NW Spitsbergen),
- Sonobuoy measurements (80 stations),
- Magnetic isochrons/picks,
- Plate boundaries
- Seafloor of heat flow.

Appendix

1. Data_point.txt

2. M_point.m
3. M_point.kml
4. M_point_imge.m
5. M_point_imge.kml
6. M_point_all.m
7. M_point_ all.kml

8. P_point.py
9. P_point.kml
10. P_point_imge.py
11. P_point_imge.kml
12. P_point_all.py
13. P_point_all.kml

14. Data_line.txt

15. M_line.m
16. M_line.kml
17. M_line_imge.m
18. M_line_imge.kml
19. M_line_all.m
20. M_line_all.kml

21. P_line.py
22. P_line.kml
23. P_line_imge.py
24. P_line_imge.kml
25. P_line_all.py
26. P_line_all.kml

View publication stats