A Standard Format to Describe Earth Models and Improve Their Dissemination

Peter Danecek, Luca Postpischl, Andrea Morelli

danecek@bo.ingv.it, postpischl@bo.ingv.it, morelli@bo.ingv.it

Istituto Nazionale di Geofisica e Vulcanologia - Sezione di Bologna, Via Donato Creti 12, 40128 Bologna, Italy

2008 AGU FALL MEETING
15-19 DECEMBER · SAN FRANCISCO
IN33A-1159

RATIONALE

Earth models, resulting from seismic tomographic studies, are expressed following very different conventions and formats. A potential user must check format descriptions, and often run or code computer scripts and programs before being able just to display the models. This fact limits diffusion and use of tomographic results.

A standard representation for the exchange and distribution of earth models would enhance the circulation and application of models both by fellow seismologists, and by a broader non-specialist community. Such a representation will not need to be used by tomographers in the calculation and data modeling stages, but only to define an interexchange format. This subject has been given consideration within the Research Activity dedicated to defining a European reference earth model in the EU NERIES research project [1].

SEMANTIC DATA STRUCTURES AND JSON

Typical datasets for grid-based tomographic earth models consist of **large quantities of data**, while the semantic approach to data representation is an important feature for building flexible software agents, we find that the standards-based XML approach typically results in unnecessary large files with a low scientific content to mark-up ratio.
We thus looked for alternative solutions capable of being **language-independent**, easy to **parse**, **semantic**, i.e. resulting in self-describing structures integrating data and metadata (authors, institutions, bibliographic references, units of measure, Reference System) in a single resource that can be recognized, **indexed** and efficiently **processed** both in web contexts and in high-level scientific programming environments.

We believe the Json formalism fits these general requirements well and we propose to adopt it for the standardization of earth models. **Json** (“Javascript Object Notation” [2]) is a subset of the ECMA-262 specification; it is based on a very minimal and clean notation, that should be already familiar to most programmers, given its derivation from basic structures of C/C++. As such, it implements a formally-strict **syntax**, which serves as a solid base to simplify **parsing** routines.

Compared with an equivalent XML implementation a **json** object is significantly light-weight in file size (20-30% less); moreover **json** formatted structures are fully defined **Javascript** objects, as such their elements are directly accessible by scripts, without the typical DOM-traversing parsing routines needed for XML. **Json** is thus very **efficient** and it is becoming the preferred data-exchange format in many **Web 2.0** contexts.
Json fits by default with the open-source frameworks developed in the Web Standards Community, adopting it thus means having a huge arsenal of software tools readily available to build advanced, client-side interfaces to the data, within the web browser.

On the other hand there are now **json** libraries or built-in **json** support for a growing list of programming languages amongst which ActionScript, C, C++, C#, Cold Fusion, D, Delphi, E, Erlang, Haskell, Java, Lisp, LotusScript, Lua, Perl, Objective-C, OCAML, PHP, Python, Rebol, Ruby, Scheme, and Squeak; thus conversions toward higher levels formats for scientific data such as netCDF and HDF5 are easy to implement [3], allowing advanced plotting with visualization tools such as the GEON IDV [3] and GMT[4].

json is not a document format, nor a markup language, it is just a notation convention to represent name-value pairs; it is thus a very flexible mean, capable of including new elements and variables with no backward compatibility issues.

IMPLEMENTATION

A seismic tomographic model is often represented by its **values on a 3D grid**. The current proposed JSON implementation strives to exploit grids’ regularities to keep the resulting file compact. The notation for structured grids takes advantage of regular spacing in any direction and only specifies the incremental step for it, either as a constant or as a function defined iteratively.
When the incremental step variable is set to ‘null’ the explicit enumeration of the values as a standard Javascript array is required, this gives support for irregular grids.
The sets of depth, latitude and longitude values for grids are represented as separate sub-entities of the main Json object, this **avoids unnecessary repetitions of coordinates** typically found in (lat, long, depth)-triplets implementations. The resulting format is flexible enough to freely intermix the modalities for depth, longitude and latitude values; up to a grid where the latitude and longitude values are different on each of the depth levels provided by the model.
Another important aspect is the ability to include in the same file the values of more than one measurable, this feature can also be exploited to provide uncertainties.
An extension of the format for volumetric cells of any shape, regularly or randomly distributed in the earth interior, can be implemented with ‘FeatureCollection’ entities, as defined in the **GeoJSON** standardization project [5].

```
{
"filetype":"Tomographic Earth Model",
"version":"0.5beta",
    "model":"s20rts",
    "authors":[
        {
            "name":"Jeroen",
            "surname":"Ritsema",
            "institution":"Geological Sciences Dept.,University of Michigan, Ann Arbor, Usa"
        },
        {
            "name":"H. J.",
            "surname":"Van Heijst",
            "institution":"Geological Sciences Dept.,University of Michigan, Ann Arbor, Usa"
        }

    ],
    "doi":"10.1089/ast.2008.0759",
    "title":"Seismic imaging of structural heterogeneity in Earth's mantle: Evidence for large-scale mantle flow",
    "format":"pointdata",
    "dataset":{
        "mode":"compact",
        "mapping":"latlon",
         "crs":"WGS84/sphere",
        "depth":{
            "min":50,
            "max":1600,
            "unit": "km",
            "step": null,
            "values":[50,100,150,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500,1600]
        },
        "lat":{
            "min":30,
            "max":90,
            "unit": "degree/radiant",
            "step": 2,
            "values": null              //not used when 'step' is set, useful for irregular latlon grids
        },
        "lon":{
            "min":-40,
            "max":60,
            "unit": "degree/radiant",
            "step":2,
            "values":null
        },
        "measurables":[
            {
            "name":"s-wave velocity",
            "abbr": "Vs",
            "units":"km/s",
            "error":null,
            "values":[

[4.42817,4.42531,4.42955,4.44069,4.45816,4.48067,4.50592,4.53043,4.54990,4.56013,4.55824,4.54378,4.51933,4.49014,4.4630
0, ...],
[,4.62573,4.62782,4.62329,4.61532,4.60867,4.60784,4.61498,4.62852,4.64323,4.65182,4.64748,4.62656,4.59013,4.54402,4.497
11, ..],
[4.35141,4.34740,4.35541,4.37531,4.40519,4.44141,4.47890,4.51183,4.53473,4.54365,4.53726,4.51742,4.48905,4.45917,4.4353
7, ...],
[4.70834,4.70089,4.69041,4.68120,4.67721,4.68078,4.69164,4.70655,4.71982,4.72485,4.71603,4.69061,4.64981,4.59872,4.5450
2, ...],
[4.32506,4.31393,4.31035,4.31658,4.33338,4.35928,4.39043,4.42124,4.44574,4.45919,4.45950,4.44793,4.42890,4.40889,4.3948
0,, ..],
[4.71832,4.70246,4.68486,4.67115,4.66528,4.66852,4.67911,4.69266,4.70326,4.70503,4.69359,4.66724,4.62744,4.57848,4.5265
4, ...]
...
[6.72858,6.72523,6.72727,6.73431,6.74456,6.75542,6.76430,6.76943,6.77022,6.76728,6.76195,6.75581,6.75007,6.74540,6.7419
1, ...]                                          ]
            }
        ]
    }
}
```

The Json standardization format proposal displayed in the sidebox is also available at http://www.bo.ingv.it/eurorem with all related documentation.

Once a json object is available on a web page and assigned to a generic javascript variable like for example 'sampleRemModel', all 'properties' of the object can be accessed directly by the 'dot-notation':

```
sampleRemModel.doi;
 //returns the metadata field containing the doi number

sampleRemModel.authors.length;
 //returns the number of subitems found in the 'authors' property
    (2 in the example above)

sampleRemModel.dataset.lat.max;
 //returns 90

sampleRemModel.dataset.measurables[o].values[m][n]
 //returns the n-th element of the sub-array of 'measurables'
    containing the values for the 'm-1' depth.
```

The **json** data are thus readily available to be analyzed within the browser and processed with any of the advanced javascript frameworks freely available.

json parsing is equally easy in PHP, which provides a dedicated native function, json_decode.

Given PHP capabilities in setting protocol headers and creating files on the server-side, it is the best choice to export the json data in other formats and automatically launch the right application for them.

```
<?php
$json = '{"filetype":"Tomographic Earth Model",
          "version":"0.5beta",
          "model":"s20rts"
            (...)
          }';

$obj = json_decode($json);
print $obj->{'title'};  // prints: Seismic imaging of structural...
(...)
   ?>
```

TOOLS

A sample Javascript visualization and comparison tool for earth models is available, as a proof-of-concept, this plots the frequency distributions of velocities values directly from the Json data for over 20 Earth Models. Using the same Javascript methods based on the HTML ‘<canvas>’ tag, a geographic map can be drawn directly on the HTML page, and quick conversions towards other formats provided.
Only the web-browser client-side resources are needed, thus we plan to evolve this example into a general tool to be distributed as a simple HTML/Javascript file. Without any installation scientist will be able to locate a Json-formatted earth model either locally or by specifying a URL, and visualize or export the data.

We also developed another sample Json processing routine written in **PHP** which plots the set of reviewed Earth Models in GoogleEarth. **KML** [6] (the XML flavor used in this context) conversion of single cells coordinates from the json data is straightforward but over-verbose; thus we build the main file as a hierarchy of nested KML folders for the models and their depths, where each folders contains a '<NetworkLink>' that retrieves data from the remote php script only when the user explicitly requests to view that particular layer, by clicking on the corresponding title in GoogleEarth sidebar.
The newly released Google Earth plugin for web browsers and the associated Javascript API is also being evaluated to merge these two sample applications into the final tool.

CALL FOR FEEDBACK

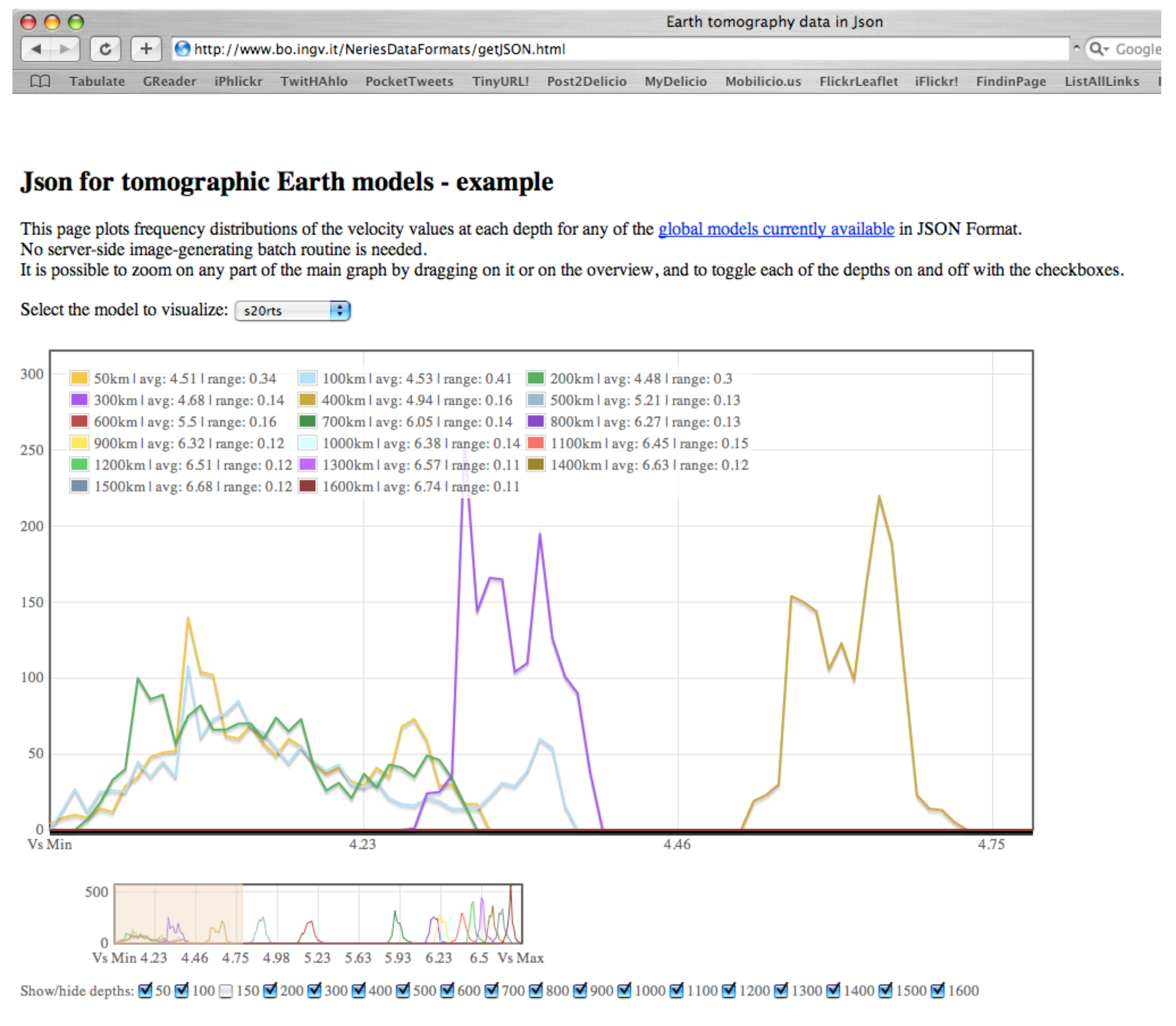
All revisions of the format will be discussed and implemented openly on the wiki page http://www.bo.ingv.it/wiki/index.php/JsonREM. .We invite all interested scientists to contribute to the discussion. In particular, the **inclusion of the spherical harmonics coefficients** in the exchange format is to be considered, given that most global earth models in literature had been expressed in such mathematical terms, and that software packages based on them exist.

ACKNOWLEDGEMENTS

We thank Simona Carannante for useful suggestions. Work supported by EC FP6 I3 project NERIES (RII3-CT-2006-026130).

REFERENCES

[1] http://www.neries-eu.org
[2] htpp://www.json.org
[3] http://geon.unavco.org/unavco/IDV_for_GEON_data_converters.html
[4] http://gmt.soest.hawaii.edu/
[5] htp://www.geojson.org
[6] http://code.google.com/apis/kml/


Earth tomography data in Json
http://www.bo.ingv.it/NeriesDataFormats/getJSON.html
Tabulate GReader iPhlickr TwitHAhlo PocketTweets TinyURL! Post2Delicio MyDelicio Mobilicio.us FlickrLeaflet iFlickr! FindinPage ListAllLinks
300
250
200
150
100
50
0
Vs Min
4.23
4.46
4.75
500
0
Vs Min 4.23 4.46 4.75 4.98 5.23 5.63 5.93 6.23 6.5 Vs Max

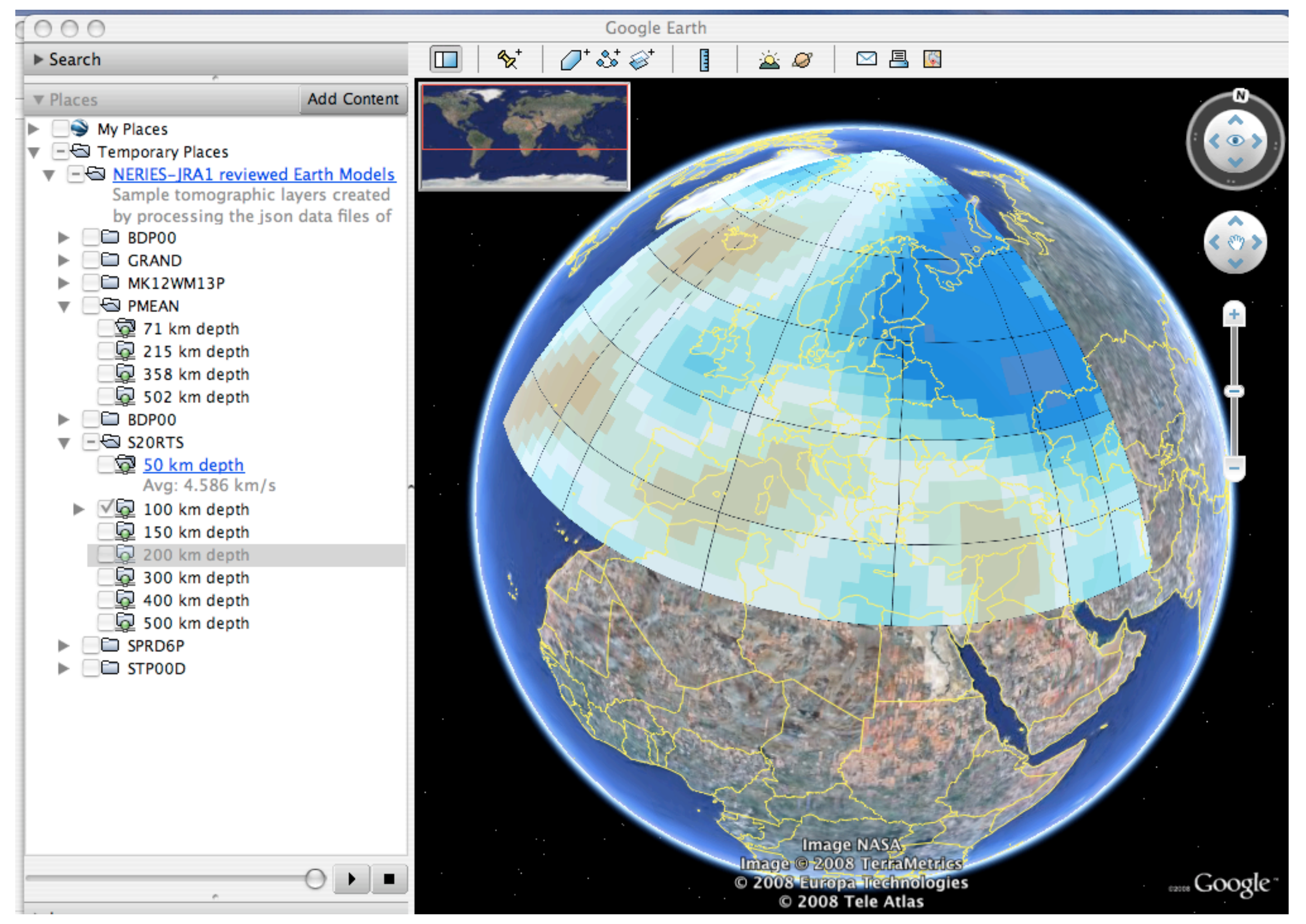
Json for tomographic Earth models - example

This page plots frequency distributions of the velocity values at each depth for any of the global models currently available in JSON Format.
No server-side image-generating batch routine is needed.
It is possible to zoom on any part of the main graph by dragging on it or on the overview, and to toggle each of the depths on and off with the checkboxes.

Select the model to visualize: s20rts

50km | avg: 4.51 | range: 0.34
100km | avg: 4.53 | range: 0.41
200km | avg: 4.48 | range: 0.3
300km | avg: 4.68 | range: 0.14
400km | avg: 4.94 | range: 0.16
500km | avg: 5.21 | range: 0.13
600km | avg: 5.5 | range: 0.16
700km | avg: 6.05 | range: 0.14
800km | avg: 6.27 | range: 0.13
900km | avg: 6.32 | range: 0.12
1000km | avg: 6.38 | range: 0.14
1100km | avg: 6.45 | range: 0.15
1200km | avg: 6.51 | range: 0.12
1300km | avg: 6.57 | range: 0.11
1400km | avg: 6.63 | range: 0.12
1500km | avg: 6.68 | range: 0.12
1600km | avg: 6.74 | range: 0.11

Show/hide depths: ✓ 50 ✓ 100 ☐ 150 ✓ 200 ✓ 300 ✓ 400 ✓ 500 ✓ 600 ✓ 700 ✓ 800 ✓ 900 ✓ 1000 ✓ 1100 ✓ 1200 ✓ 1300 ✓ 1400 ✓ 1500 ✓ 1600


Google Earth
Search
Places
Add Content
My Places
Temporary Places
NERIES-JRA1 reviewed Earth Models
Sample tomographic layers created by processing the json data files of
BDP00
GRAND
MK12WM13P
PMEAN
71 km depth
215 km depth
358 km depth
502 km depth
BDP00
S20RTS
50 km depth
Avg: 4.586 km/s
100 km depth
150 km depth
200 km depth
300 km depth
400 km depth
500 km depth
SPRD6P
STP00D
Image NASA
Image © 2008 TerraMetrics
© 2008 Europa Technologies
© 2008 Tele Atlas
Google