20.12.2023

Internal progress report – “Wormpicker” and data ingestion

1. Project goal

One of the major objectives of this Use Case is to conceive and implement an advanced computational workflow facilitating the efficient handling of both genomic and environmental data within data cubes.

From rasterized data to point estimates

The genomic data used for this Use Case was generated from *Drosophila melanogaster* samples collected in natural populations. Prior to sequencing, 10-40 flies from each location were pooled for DNA extraction and the resulting DNA pools were then sequenced with modern high throughput sequencing. These data thus do not allow to obtain genomic information from a single individual but rather allele frequency estimates for the whole population collected in nature. Since the genomic data in this project was obtained from point localities collected across different timepoints, we developed a light weighted tool that allows to access large gridded and rasterized data hosted on a server and retrieve point estimates from these data layers for a predefined set of coordinates. Linking the genomic dataset with environment data based on spatiotemporal coordinates allows to investigate the influence of the environment on genetic variation.

Since many of the analysis steps for our workflow were not possible to conduct in a hub yet, we established a functional python-based program, which can extract earth observation data provided by the project partner platform for these given coordinates and is called “The Wormpicker”. Establishing and using the Wormpicker program was necessary to allow, as mentioned before, refining the downstream analysis of the whole workflow.

2. Achievements

Wormpicker Core Functionality

The “Wormpicker” program first obtains information about available data layers on a host server and then provides a simple summary with metadata information on the earth observation data available on the respective platform. Upon authentication with valid credentials, users can choose the layers data that should be retrieved. Once the workflow is completed, these layers are processed and point estimates for the input coordinates will be available for follow-up analyses. To streamline data retrieval, core functionalities gather the necessary information on the platform in order to formulate correct queries in WCPS language to locate the data on the database. Specifically, the script processes XML files provided by the web-based server and navigates through the system-specific architecture to obtain meta-information from each layer, which include the coordinate reference system, geographic extent, time range etc. The program then uses all this information to correctly calculate grid indices, which correspond to the desired coordinates and timepoints, for which data should be retrieved. The initial setup of the program was designed to send queries by directly addressing coordinates. Due to inconsistencies in grid index calculation by rasdaman, we were offered a solution to work around this platform inherited issue and query grid indices directly. From these grid indices, queries are formulated for retrieving data from a specific point in space and time, which correspond to the coordinates of the sampling localities of *Drosophila melanogaster* populations. Practically these points can be any localities of interest. If data for these points is available, these are exported to a csv-file.

At the moment, we have reached a development stage, where data retrieval as described above is possible with the Wormpicker. However, several server-side issues, as described below, hinder us from continuing the development of this program.

3. Issue

Wormpicker

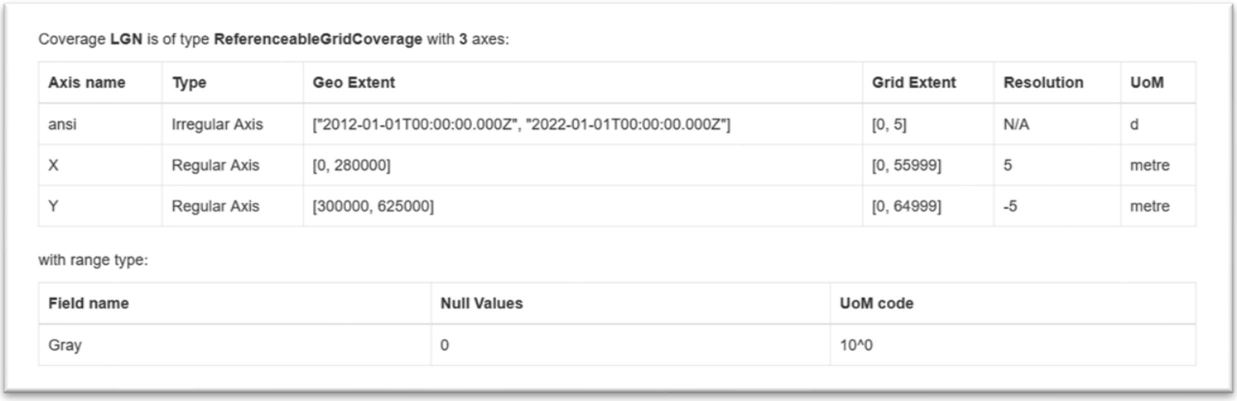
Since the first stable draft of the program, there have been several occasions where the Wormpicker stopped working properly and data could not be retrieved in a reliable manner. Upon continuously testing our code for bugs, we believe that the reason for this behaviour was rather changes in the platform architecture at rasdaman. We found that certain parameters and information that are required by the Wormpicker to operate were no longer available. Since then, we have encountered many instances of undocumented structural changes that broke the Wormpicker functionality to reliably access platform architecture and to obtain information. Unfortunately, since no documentation is available for these unplanned changes, it is hard for us to assess whether these structural changes follow an intentional scheme, which makes it difficult for us to apply changes in the code to account for these changes. Furthermore, we found very inconsistent use of data standards and naming conventions, which makes it currently impossible for us to implement standardized procedures in our program to process the meta data on rasdaman. Please find below a few examples:

These inconsistencies are, amongst others, reflected in the following structural components:

a) Axis Labelling/Axis Name: These differ across layers and it is not clear or documented which naming convention they follow. We understand that this originates from different CRS labelling conventions originally, anyhow there are inconsistencies especially in the time dimension, being referred to as “ansi”, “time” or “date” independent from the CRS used.

b) Geo Extent: Across different layers, time geo extent (respectively time extent or date extent) has different formats. On axes “X” and “Y”, for the layer “Near Surface Air Temperature” we encounter coordinate data, that cannot be handled by many geo-informatic functions implemented in various programs or python packages.

Examples

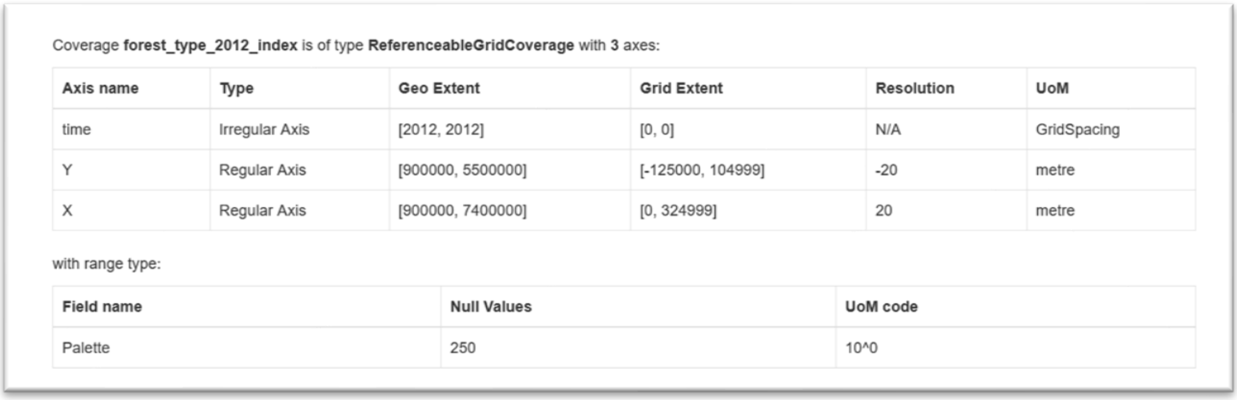


Coverage **LGN** is of type **ReferenceableGridCoverage** with **3** axes:

Axis name	Type	Geo Extent	Grid Extent	Resolution	UoM
ansi	Irregular Axis	["2012-01-01T00:00:00.000Z", "2022-01-01T00:00:00.000Z"]	[0, 5]	N/A	d
X	Regular Axis	[0, 280000]	[0, 55999]	5	metre
Y	Regular Axis	[300000, 625000]	[0, 64999]	-5	metre

with range type:

Field name	Null Values	UoM code
Gray	0	10^0

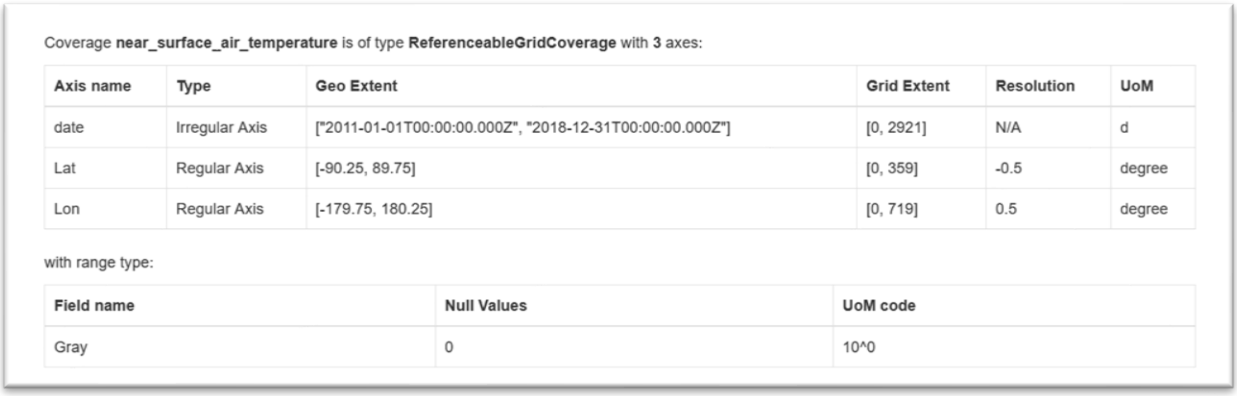


Coverage **forest_type_2012_index** is of type **ReferenceableGridCoverage** with **3** axes:

Axis name	Type	Geo Extent	Grid Extent	Resolution	UoM
time	Irregular Axis	[2012, 2012]	[0, 0]	N/A	GridSpacing
Y	Regular Axis	[900000, 5500000]	[-125000, 104999]	-20	metre
X	Regular Axis	[900000, 7400000]	[0, 324999]	20	metre

with range type:

Field name	Null Values	UoM code
Palette	250	10^0

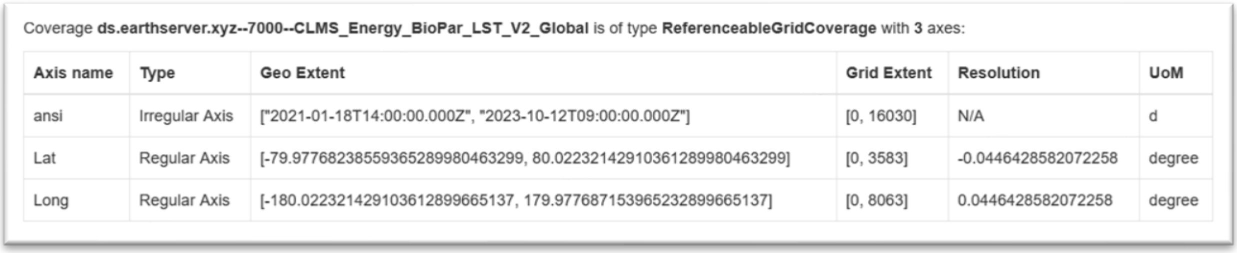


Coverage **near_surface_air_temperature** is of type **ReferenceableGridCoverage** with **3** axes:

Axis name	Type	Geo Extent	Grid Extent	Resolution	UoM
date	Irregular Axis	["2011-01-01T00:00:00.000Z", "2018-12-31T00:00:00.000Z"]	[0, 2921]	N/A	d
Lat	Regular Axis	[-90.25, 89.75]	[0, 359]	-0.5	degree
Lon	Regular Axis	[-179.75, 180.25]	[0, 719]	0.5	degree

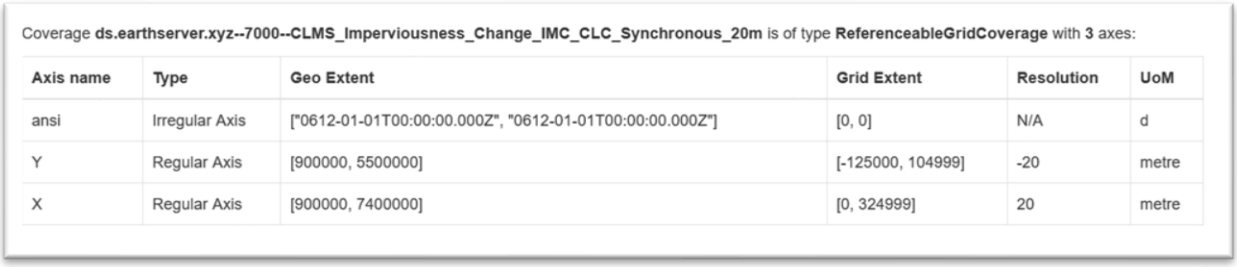
with range type:

Field name	Null Values	UoM code
Gray	0	10^0



Coverage **ds.earthserver.xyz--7000--CLMS_Energy_BioPar_LST_V2_Global** is of type **ReferenceableGridCoverage** with **3** axes:

Axis name	Type	Geo Extent	Grid Extent	Resolution	UoM
ansi	Irregular Axis	["2021-01-18T14:00:00.000Z", "2023-10-12T09:00:00.000Z"]	[0, 16030]	N/A	d
Lat	Regular Axis	[-79.97768238559365289980463299, 80.02232142910361289980463299]	[0, 3583]	-0.0446428582072258	degree
Long	Regular Axis	[-180.022321429103612899665137, 179.977687153965232899665137]	[0, 8063]	0.0446428582072258	degree



Coverage **ds.earthserver.xyz--7000--CLMS_Imperviousness_Change_IMC_CLC_Synchronous_20m** is of type **ReferenceableGridCoverage** with **3** axes:

Axis name	Type	Geo Extent	Grid Extent	Resolution	UoM
ansi	Irregular Axis	["0612-01-01T00:00:00.000Z", "0612-01-01T00:00:00.000Z"]	[0, 0]	N/A	d
Y	Regular Axis	[900000, 5500000]	[-125000, 104999]	-20	metre
X	Regular Axis	[900000, 7400000]	[0, 324999]	20	metre

c) Location of information in the XML and namespaces: Layers do not uniformly follow the same namespace patterns and therefore exception handling statements are constantly incorporated in the Wormpicker core functionalities. As non-experts it is not clear to us if the ones we encounter on the platform are all possible namespaces and if they follow a logic that is standardised and useable across different OGC services (public and private). There are two “Versions” of WCS available (WCS 2.0.1 and 2.1.0). This is also reflected in the XML name spacing by using version names for certain XML elements, differing among layers. The same concerns the coverage types “CIS 1.0” and “CIS1.1” with the respectively different Grid Coverage Types getting addressed. It is unclear what differentiates these two versions of WCS and the Grid Coverage Types and how their properties influence our program and workflow, as well as where or how an explanatory statement or notification has been communicated during these processes of change.

d) Information about values (Metadata): At the moment it is still not clear where to extract information about the actual meaning and context of the values that are represented by a layer. For us, many of the ingested layers are not usable, due to the fact that we do not have the necessary information concerning source of the data, ownership of the data and what this data is meant to represent. An interpretation cannot be concluded from a layer name reliably. It was possible to follow up metadata discussions on GitHub for one layer which is “Near Surface Air Temperature”. We could conclude, that the producer and provider of the data (Copernicus) defines the measure as “Temperature in Kelvin”. Therefore, we accessed this layer with the Wormpicker to retrieve data for our points of interest. We are not aware of metadata standards on the platform. Unfortunately, other layers have no clear structure concerning the ingestion process and manually created issues on GitHub partially intersect with automatically generated issues based on an inventory sheet. The lack of a uniformly structured source of metadata is preventing a smooth integration of this necessary information in a time efficient manner. This concerns the Earth Server Federation data. Analytical usage of corresponding layers is not possible due to lack of essential information that is needed to correctly use, interpret and cite the data.

In summary, it would be essential to be informed a priori about fundamental changes in the rasdaman architecture and to have access to a well-document changelog, so that we can adjust our program timely and in accordance with the documented updates. So far, we had to idiosyncratically adjust our program by trial and error to restore its functionality. Nevertheless, as non-experts in the field of earth observation science and database informatics, we, have to devote a large amount of time and capacities into fixing these errors and we would be very glad if we could rather invest our resources into the core objectives of our use case, i.e., to develop data analysis strategies and to address biological questions.

There is more need for support in these infrastructural aspects that have been accessible to us until now. Considering the aspect, that aside from querying data, also processing of the results is an essential part of the platform and project goal, we also want to address that there has not been progress on the processing of data on the platform yet. All the use case specific analysis of data was conducted locally so far.

Ingestion of genomic and earth observation data

Earth observation data

The layer data requested by our use case are to a very large extent overlapping with requests from the other use cases. Our main focus lies on environmental measurements like temperature, humidity, and precipitation which may represent important ecological factors. Furthermore, we want to include data on the anthropogenic influence on environmental conditions like demography, land use or pesticide application. We now added additional information on our priority ranking of these data layers to the Ingestion issues on GitHub. From our list of requested data layers, only “surface air temperature” have been ingested to date.

All the requested data layers are free and publicly available and can be accessed online. Data availability is therefore not the limiting factor in the desired research field. Since we are no experts in the field of earth observation science we are very grateful for any support with, e.g., data standards in metadata information, etc. Moreover, we are thus not able to provide expert information on data standards required for ingestion.

Genomic data

A draft on how to ingest data was made in the FAIRiCUBE meeting in Bremen in 2023 and access to the whole dataset for the genomic data of *Drosophila melanogaster* was shared, as well as a gapless and purified subset of this data in the form of a VCF, called “DrosoSandbox”, was provided. This ingestion of genomic data has seen minor progress so far and requires more dedicated work from all the responsible parties. Given the unresolved issues raised above, we do not prioritise the cubification of genomic data at the moment.

Communicating these issues

A lot of these aspects have been addressed multiple times during meetings and in several GitHub issues, as well as via E-Mail conversations. As biologists and non-experts in the field of earth observation, aside from self-conducted research and investigation, it is essential for us to keep conversation and discussion ongoing to facilitate gain of knowledge and better understanding of certain domain specific infrastructures.

4. Conclusion

The initial phase of the project has yielded achievements, reaching small goals towards the realization of the proposed workflow. The Use Case team successfully implemented the first draft of a software designed for querying point data. However, the aforementioned problems and issues concerning the platform structure which results in continued adjustments of the Wormpicker to restore functionality, has led us to the decision to stop conducting work in the software development aspect at the moment until we find a solution for the issues mentioned above. Until now, we have successfully applied to Wormpicker approach to obtain data from the “near surface air temperature” layer, which is the only available and informative dataset at the moment. We will for now use this dataset to focus on the landscape genomics objective of our use case.

Since there was no progress in the ingestion of genomic data, we decided to not invest into the development of user defined functions (UDF) for a server-side analysis of our dataset but rather conduct these data locally on the NHM systems. Simultaneously the FAIRiCUBE Hub (operated by EOX) will be investigated further in terms of suitability for intersection of earth observation data with *Drosophila melanogaster* genomic data as well as downstream analysis of such results.

In this report we wanted to document our progress and unresolved issues in the context of our project goals. We hope that this document will help to identify and resolve these unresolved issues and to foster communication.