DeepMIP database overview

This information provides an overview of the EARTHSEQUENCING database structure, and how to provide new data for the DeepMIP effort.

General overview:

The database system uses Couchbase as the backend Server technology, allowing much faster access than traditional relational databases, yet preserving the typical cross-table relationships that allow relating, e.g., core-depths and sample identifiers to derived properties like age models. The idea is that this backend stores the relevant data and meta-data, and eventually allows programmatic and external access to data and derived properties.

In order to avoid the slow-access and cumbersome retrieval of data from, e.g., the IODP LIMS system, where each individual measurement is stored in its own row, the EARTHSEQUENCING database stores data aggregated on a *per-Hole* and *per-Analysis/Datatype* basis, with supporting structural meta-data, formatted as MessagePack and heavily compressed using the ZSTD algorithm. Overall this approach results in access times per hole and analysis on the order of seconds, rather than minutes as with LIMS. The actual data tables are flexible, and can be imported from traditional TSV (tab separated, preferred) or CSV files that can be exported from all major spreadsheet applications.

Proof-of-concept demo site

To get a general idea of the what the data tables might look like in our system, we have currently a proof-of-concept site, publicly accessible at https://paloz.marum.de. This site requires Javascript to be enabled (default on most browsers).

Please note that this will not be the final system, and furthermore does not demonstrate our workflow system that is supposed to support on-the-fly age model and depth splicing capabilities (and much more). The hooks to programmatically retrieve and search data are also not currently exposed publically, but it should be enough to give you a rough idea of how different kinds of data are imported.

Also, the data currently exposed only cover ODP 101 through IODP 346, and only for select MST style data. A full import of all data, including core images, additional isotope data, or DESC core description data is currently underway, and eventually will at least cover all of DSDP, ODP and IODP. The system is not limited to ODP data, though, and can accommodate any kind of structured data, as long as there is some kind of identifier per sample.

Below shows the initial demo page: you can filter by entering text in the empty fields, or sort.

5525 rows loaded of 5525 available. Click on row to show Analysis-Hole Details
Export to CSV available on striped button on right in Detail page.

Exp	Site	Hole	Analysis	Datarows	
101	626	A	GRA	526	CSV
101	626	B	GRA	2265	CSV
101	626	C	GRA	11339	CSV
101	626	D	GRA	1193	CSV
101	627	A	GRA	1505	CSV
101	627	B	GRA	50576	CSV
101	628	A	GRA	24578	CSV
101	629	A	GRA	502	CSV
101	630	A	GRA	2422	CSV
103	637	A	GRA	13483	CSV
103	638	B	GRA	30525	CSV
103	638	C	GRA	5092	CSV
103	639	A	GRA	4804	CSV

Total Items: 5525

There are two ways to export data: A click on the CSV button will directly download a CSV file, but due to limitations in different browsers only some will name the Download file appropriately (Safari, for example, does not). The other way is to do a row-click which will drill down and show the actual data. In the top right corner of that page (below), there is an action button ☰ which also allows exporting the data, and properly naming the generated file.

Back to Summary List

expedition	site	hole	core	core_type	section	offset_top...	depth_mbsf	density
101	626	A	1	R	1	0.6	0.006	1.334
101	626	A	1	R	1	1.2	0.012	1.581
101	626	A	1	R	1	1.8	0.018	1.618
101	626	A	1	R	1	2.4	0.024	1.474
101	626	A	1	R	1	3	0.03	1.291
101	626	A	1	R	1	3.6	0.036	1.437
101	626	A	1	R	1	4.2	0.042	1.448
101	626	A	1	R	1	4.8	0.048	1.368
101	626	A	1	R	1	5.5	0.055	1.432
101	626	A	1	R	1	6.1	0.061	1.392
101	626	A	1	R	1	6.7	0.067	1.305
101	626	A	1	R	1	7.3	0.073	1.385
101	626	A	1	R	1	7.9	0.079	1.163
101	626	A	1	R	1	8.5	0.085	1.49
101	626	A	1	R	1	9.1	0.091	1.602
101	626	A	1	R	1	9.7	0.097	1.703
101	626	A	1	R	1	10.3	0.103	1.682
101	626	A	1	R	1	10.9	0.109	1.713
101	626	A	1	R	1	11.5	0.115	1.689
101	626	A	1	R	1	12.1	0.121	1.653
101	626	A	1	R	1	12.7	0.127	1.694
101	626	A	1	R	1	13.3	0.133	1.694

Total Items: 526

General instructions for preparing DeepMIP data for import into database

The database upload system maintains a number of internal conversion formats to massage the underlying CSV data into a structured format. This means ideally, the data files to be uploaded should follow a certain structure, and provide a minimal set of meta-data that describe the purpose and format for each column.

Example data files are provided below in tab separated format for actual data, and meta-data:

rsc_175_1075b.txt
rsc_175_1075b_meta.txt

Important points:

1. *Data files should be valid TSV (tab separated, preferred) or CSV (second choice) files. For floating point numbers, decimal point notation should follow EN format, i.e. a dot, not a comma (this often trips up with non-EN locales like in Germany, where a comma is used instead of a dot).*
2. *For each datafile, the following meta-information should be supplied: Analysis type (e.g. GRA, MS, d18Obenthic, RSC etc), Hole (or land section, if so, provide WGS84 coordinates as lat lon), Author/Publication info, original download url where it exists, and publication doi in the form http://dx.doi.org/10.5194/gmd-2016-127.*
3. *Data files MUST be organised per Hole (or unique land section) and SHOULD be per measurement-type (where possible).*
4. *Each column SHOULD have a unique name. That means that e.g., multiple columns with a header called "Depth" should be made unique where possible. It is possible to override this on import but means extra work.*
5. *Column header names should be all lower case, and not include spaces (can be replaced by "_" e.g.). Should be all ASCII (no things like ßÄ º etc).*
6. *Each column entry should be formatted consistently in each file as you go down each column.*
7. *Each column must relate to, and include, a specific sample identifier. For IODP, this includes for example Expedition, Site, Hole, Core, Core-type, Section, cm columns. In order to allow later calculations, this is crucial. Please split these identifiers into separate columns, as shown in the example above. Thus please not 342-U1408A-13H in one column.*
8. *Date/Time columns (e.g. for timestamps) are always difficult to import. It would help to put dates/times into the following format: "1998-09-02T14:19:00.000+0000", i.e. 4 digit year, two digits month and day, letter T, HH:mm:ss with optional fractional milliseconds followed by the time zone (+0000 for GMT). A description of date formats if you want to provide your own is given here, and the example given would correspond to "yyyy-MM-dd'T'HH:mm:ss.SSSZ".*

9. *To allow me to write proper import mappings, the following meta information also needs to be provided for each column. I think it would be best to put this into a separate CSV file with the same headers copied across, and then below the following rows*
    a. *Datatype row that gives the type of data contained in each column, selecting the quoted text from*

```
enum kNAM_Datatype : String {

case String = "string" // Any kind of text

case Double = "double" // A floating point number

case Datetime = "date" // A DateTime stamp like 1998-09-02T14:19:00+0000

case Bool = "bool" // True or False, Yes or No etc.

case Int = "int" // An Integer number

case ArrayInt = "array<int>" // Array of Ints in one column ... special use

case ArrayDouble = "array<double>" // Array of Doubles in one column  ... special use

case ArrayString = "array<string>" // Array of Strings in one column  ... special use

}
```

    b. *Semantic data meaning row, which lets the system know what kind of data is contained in each column. Exp, Site, Hole etc would be "sampleID", instrument calibration data could be "meta", specific identifiers exist for cm offset ("offset_top"), and typical IODP "depth_mbsf" and "depth_mcd" columns. It would be nice to include these for now where they exist, but these will later be computed on the fly. "value" is a generic measurement value, and if a file contains multiple measurement columns, one or several can be designated as either "value" or "value_main" or "value error", where "value_main" is the primary data type (e.g. for GRAPE: density would be "value_main", counts per second would be value). "valueerror" is for columns that provide error estimates. "-" means do not import this column into the database. "comment" is a generic comment column. Special instrument information (which device) can be marked by "instrument" or "instrument_group" but is probably only useful for IODP MST track data.*

```
enum kNAM_ImportType : String {

case SampleID = "sampleID"

case Meta = "meta"

case OffsetTop = "offset_top"

case Depth_mbsf = "depth_mbsf"

case Depth_mcd = "depth_mcd"

case Value = "value"

case ValueMain = "value_main"

case ValueError = "valueerror"

case DoNotImport = "-"

case Instrument = "instrument"

case InstrumentGroup = "instrument_group"

case Comment = "comment"

}
```

    c. *Physical unit row that provides a string describing the measurement unit, ideally in SI units. This is currently for information only, and could consist of strings like "cm", "m", "1/second", "permil VPDB", "%", "instr. units", "cie_lab", "g cm^{-3}", "ppm" etc. Have a look at the database prototype for some examples*
10. Where possible, supporting age-model or splice-table information should be provided separately (also with appropriate publication or doi or url links).

In Depth DB structure:

GeoCoreFW.xcdatamodel-2.pdf

Related articles

Content by label

There is no content with the specified labels