Tethys:
Integrating bioacoustics and habitat data

Marie A. Roch

San Diego State University

Scripps Institution of Oceanography UCSD

Bio·Waves Incorporated

Collaborators:

SIO: Simone Baumann-Pickering, Sean Herbert, Heidi Batchelor, Ana Širović, and John A. Hildebrand

NOAA NMFS: Catherine L. Berchok, Danielle Cholewiak, Lisa M. Munger, Erin M. Oleson, Denise Risch, Melissa S. Soldevilla, and Sofie M. Van Parijs

tethys.sdsu.edu

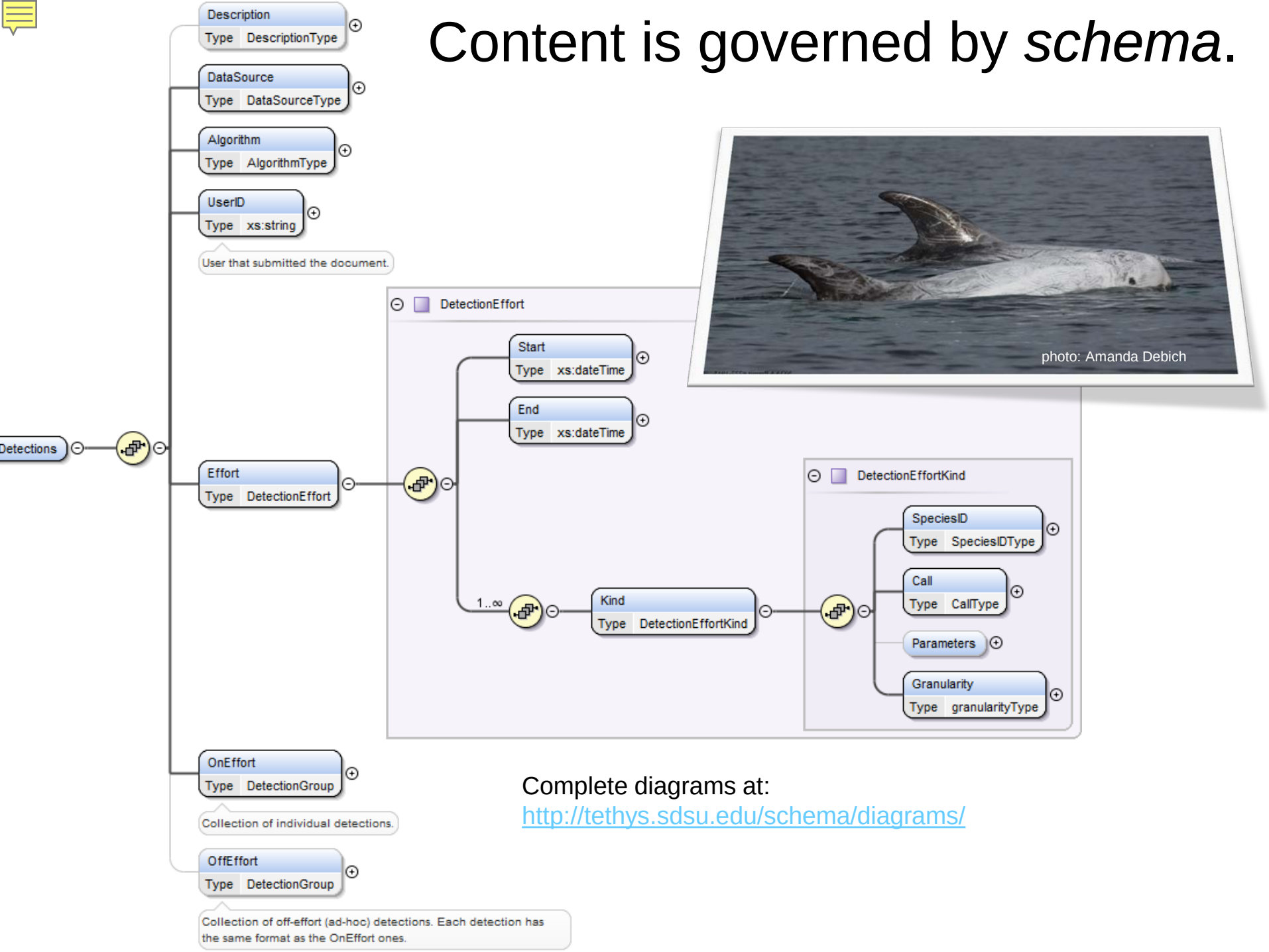
- Long-term retention of metadata
- Data standard that promotes consistency & extendibility
- Linking biological, anthropogenic and oceanographic data sets

Data Organization

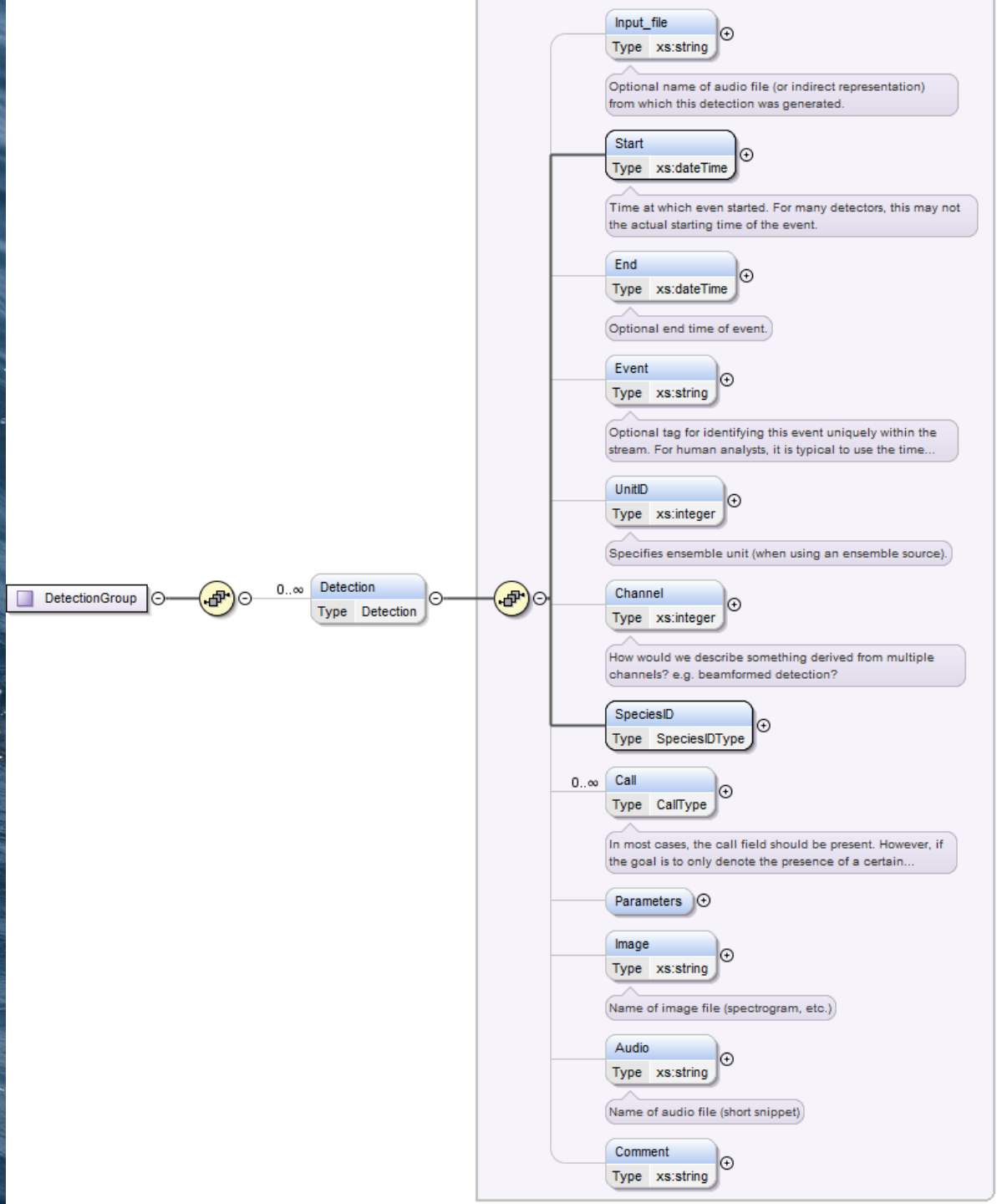
- Problem
  - Need consistency & extendibility
  - Existing standards lack critical elements
- Solution: XML

Minke whale – marinebio.org

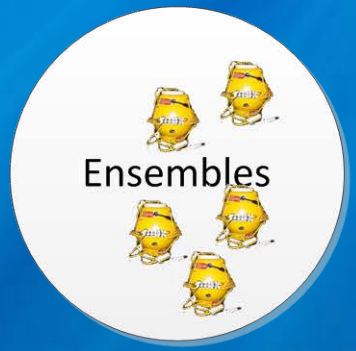
Content is governed by *schema*.



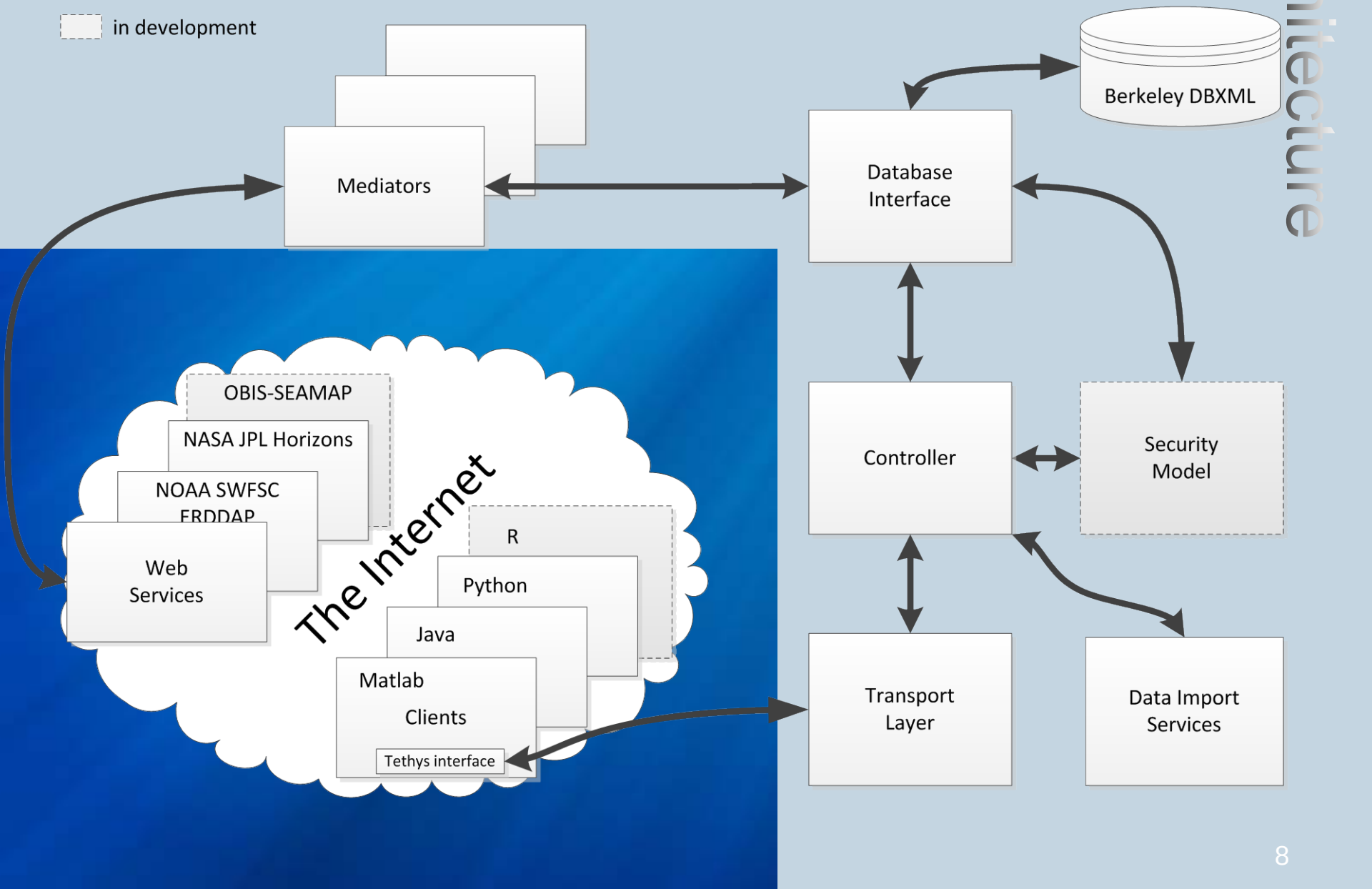
Complete diagrams at:
http://tethys.sdsu.edu/schema/diagrams/


DetectionGroup
0..∞
Detection
Type Detection
Input_file
Type xs:string
Optional name of audio file (or indirect representation) from which this detection was generated.
Start
Type xs:dateTime
Time at which even started. For many detectors, this may not the actual starting time of the event.
End
Type xs:dateTime
Optional end time of event.
Event
Type xs:string
Optional tag for identifying this event uniquely within the stream. For human analysts, it is typical to use the time...
UnitID
Type xs:integer
Specifies ensemble unit (when using an ensemble source).
Channel
Type xs:integer
How would we describe something derived from multiple channels? e.g. beamformed detection?
SpeciesID
Type SpeciesIDType
0..∞
Call
Type CallType
In most cases, the call field should be present. However, if the goal is to only denote the presence of a certain...
Parameters
Image
Type xs:string
Name of image file (spectrogram, etc.)
Audio
Type xs:string
Name of audio file (short snippet)
Comment
Type xs:string

Tethys collections



Architecture



Examination of lunar patterns and noise

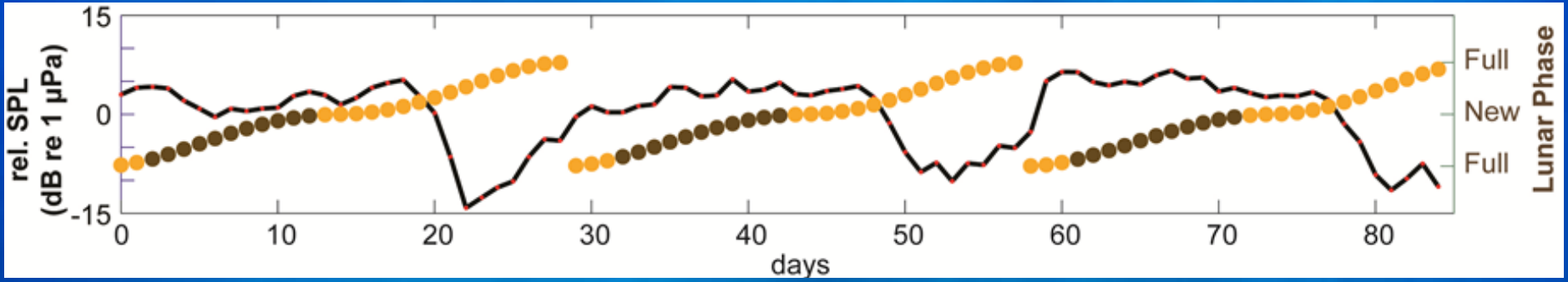
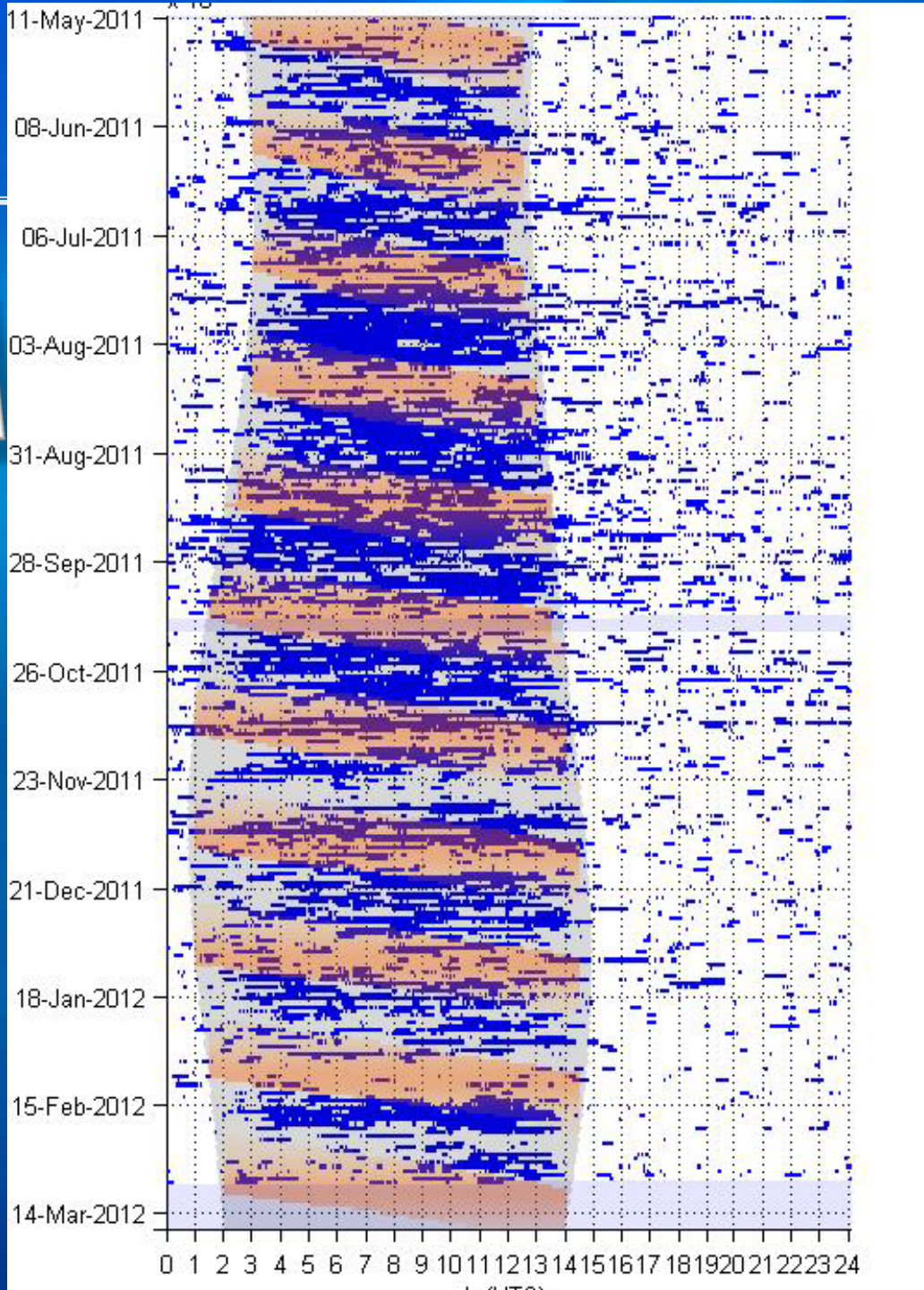
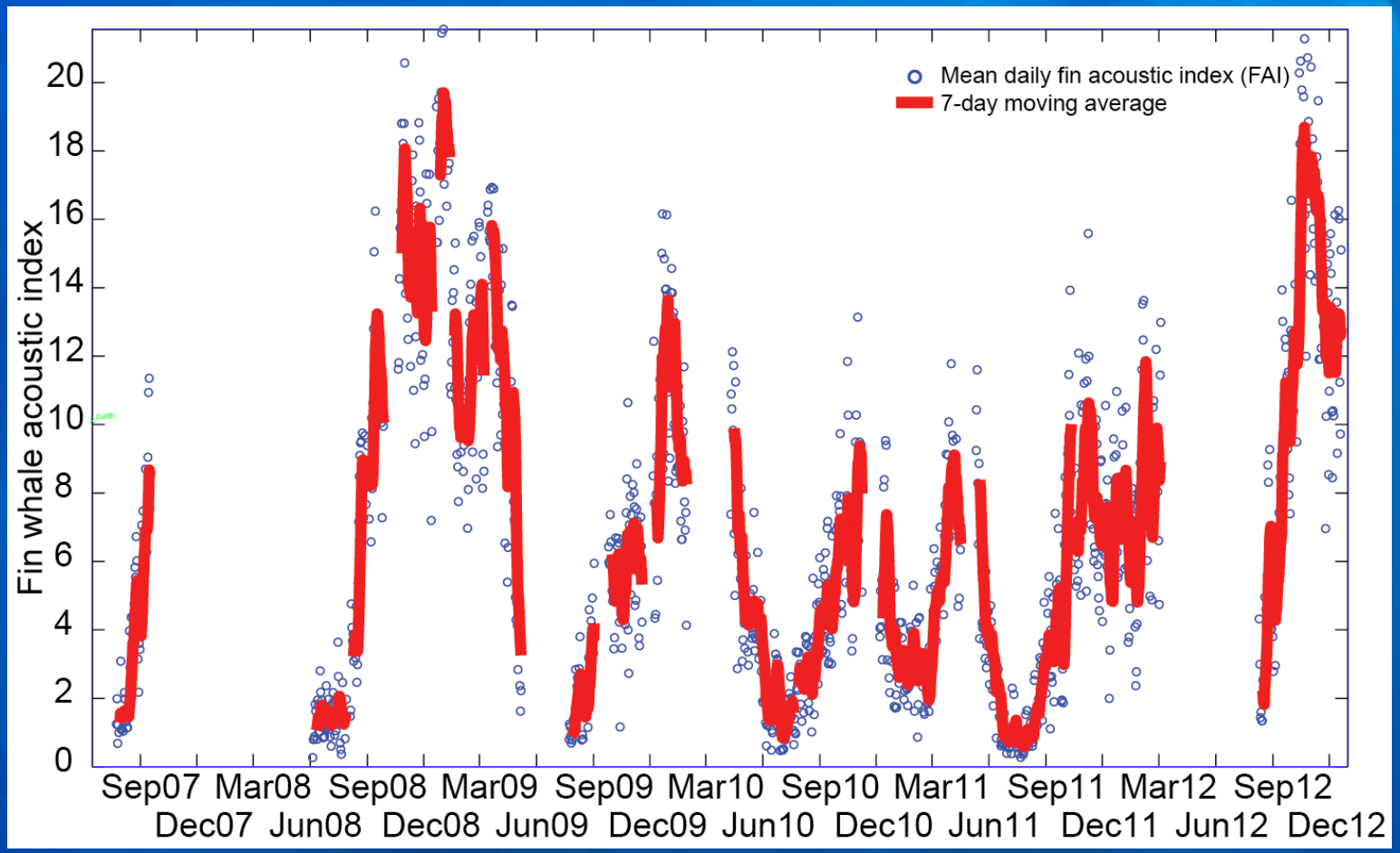


Figure 5 from our 2011 NOPP Annual Report – Coincidence of ambient noise (1-6 kHz band) and lunar phase at Wake Atoll. Visibility of moon indicated by light (visible) and dark (obscured) circles.

Lunar illumination



Longitudinal studies

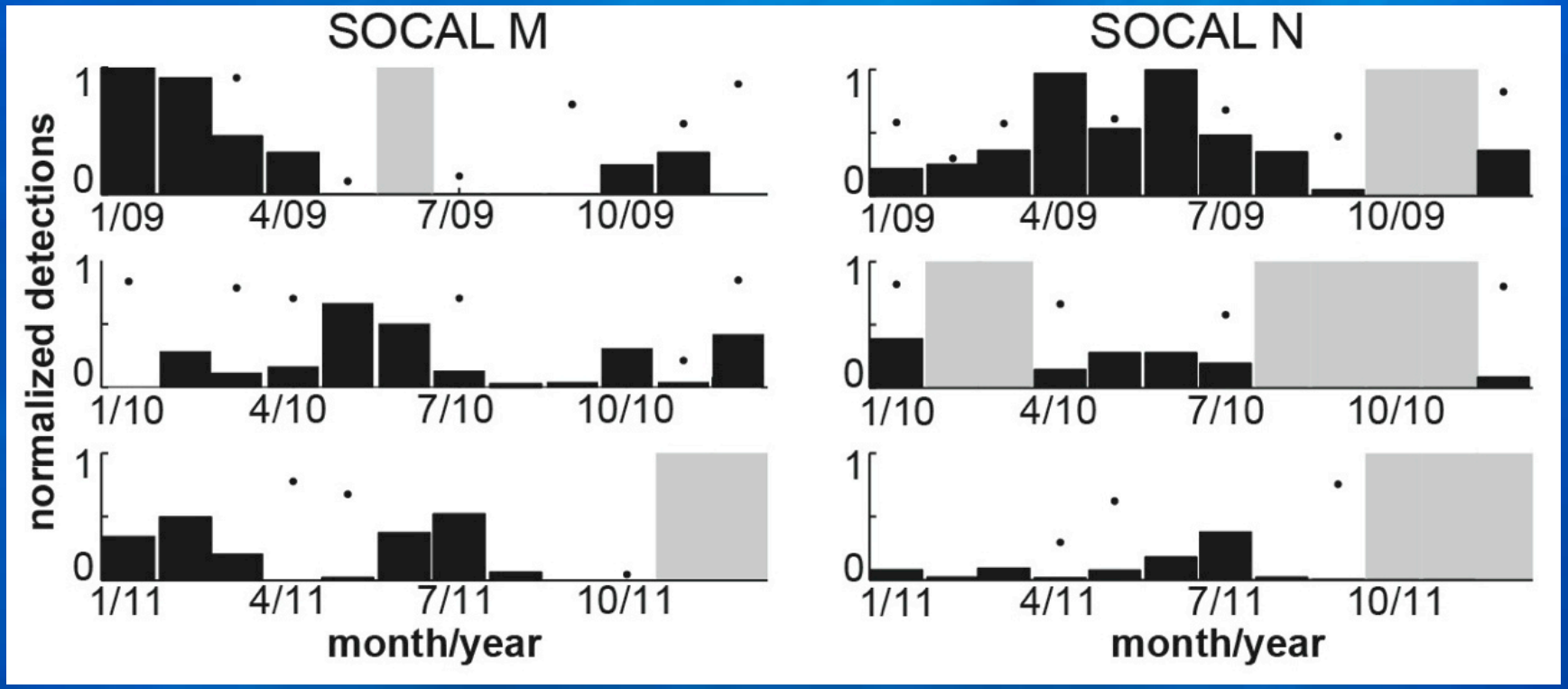


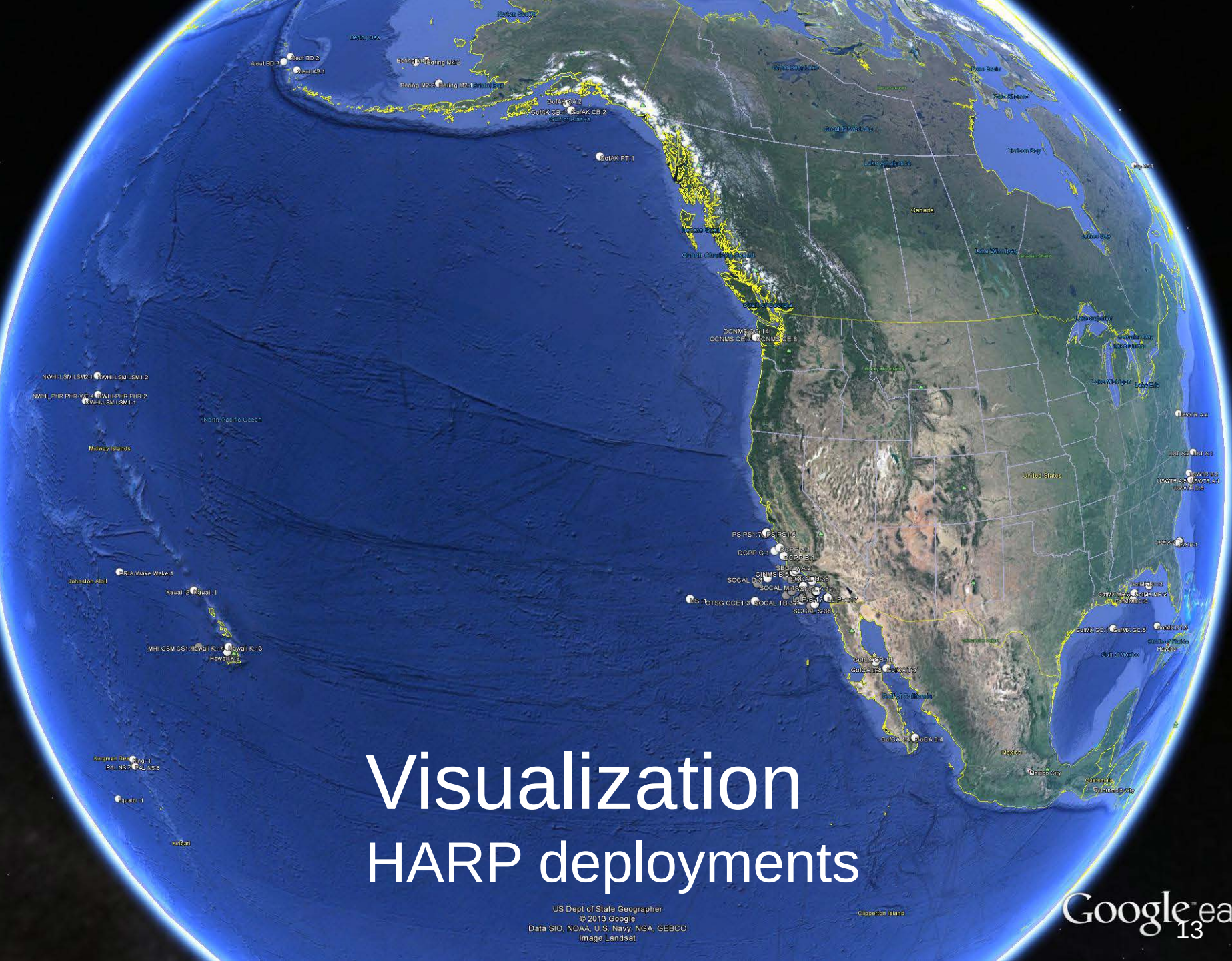
Širović et al., IBAC 2013

5 and a half years of fin whale detections

Seasonality

Cuvier's beaked whale





Visualization

HARP deployments

Tethys will…

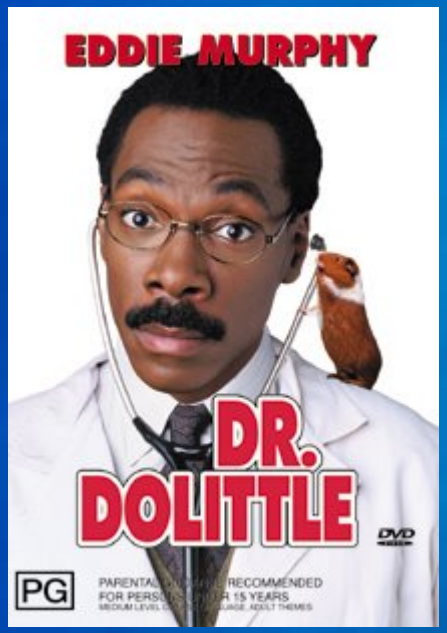
- Archive detections and localizations
- Provide access to habitat data
- Store snippets of acoustic data or images
- Provide archiving for a lab or group of collaborators

Tethys does not…

- Store large amounts of acoustic data
- Provide detection or localization routines

Nor is it designed to be a national data clearing house (e.g. OBIS-SEAMAP), but data can be exported to clearing houses.

Let's talk to Tethys



- XQuery – Query language for Tethys. Powerful. Steep learning curve
- XPath – Simplified query language. Somewhat limited.
- Client libraries
  - Python
  - Java
  - Matlab – Gentlest learning curve, richest libraries

Outline for our remaining time

- Understanding how Tethys represents metadata

- Getting our feet wet
  - Matlab Tethys queries
  - Importing data into Tethys

photo credit: Regina Hill Onyeibe

Getting started with Matlab

- Set Matlab paths

- Create a query handler

```
q = dbInit();
```

- A few useful queries:
  - dbDeploymentInfo: PAM loggers
  - dbGetEffort: What we were looking for
  - dbGetDetections: What we found

Key deployment information

- Project – Set of related deployments
- DeploymentID - #
- Site Name and/or Cruise
- Platform (mooring, tag, towed array, …)
- Instrument
- Sampling Details
- Deployment Details
- Sensors

Matlab query

```
>> d = dbDeploymentInfo(q, 'Project',
   'SOCAL', 'Site', 'H');
>> d(1) % one deployment of many
```

Project: 'SOCAL'	DeploymentID: 18
Site: 'H'	Cruise: 'Socal18'
Platform: 'mooring'	Instrument: [1x1 struct]
SamplingDetails: [1x1 struct]	DataLocation: [1x1 struct]
DeploymentDetails: [1x1 struct]	RecoveryDetails: [1x1 struct]
Sensors: [1x1 struct]	

```
>> d(1).DeploymentDetails
```

Longitude: 240.8233	Latitude: 32.8469
DepthInstrument_m: 1013	TimeStamp: '2007-07-24T00:00:00Z'
ResponsibleParty: [1x1 struct]	

Cuvier’s beaked whale effort?

```
>> [eff, effinfo] = dbGetEffort(q, 'SpeciesID', 'Ziphius cavirostris');
```

- eff – matrix of start and end times
- effinfo – Information:
  - DataSource: Deployment identifier → geolocation
  - Algorithm → tells us how we were looking
  - UserID – Who performed the analysis
  - other information…

Cuvier's South of San Clemente

San Clemente Canyon and

East Cortes Basin

32˚42'N, 118˚50'W

32˚18'N, 118˚06'W

Identifying deployments

```
d = dbDeploymentInfo(q,
    'DeploymentDetails/Latitude', {'>', 32.3},
    'DeploymentDetails/Latitude', {'<' 32.7},
    'DeploymentDetails/Longitude', {'>', 241.17},
    'DeploymentDetails/Longitude', {'<', 241.9})
```

Cuvier’s effort south of San Clem.

[eff, effInfo] = dbGetEffort(q, 'Site', {'S', 'N', 'NN', 'NW', 'NE', 'NS'}, 'SpeciesID', 'Ziphius cavirostris')

Returns

- eff - Series of date intervals
- effInfo – Array of detailed information

Cuvier’s detections S. of San Clem.

[det, endP, detinfo] = dbGetDetections(q, 'Site', {'S', 'N', 'NN', 'NW', 'NE', 'NS'}, 'SpeciesID', 'Ziphius cavirostris')

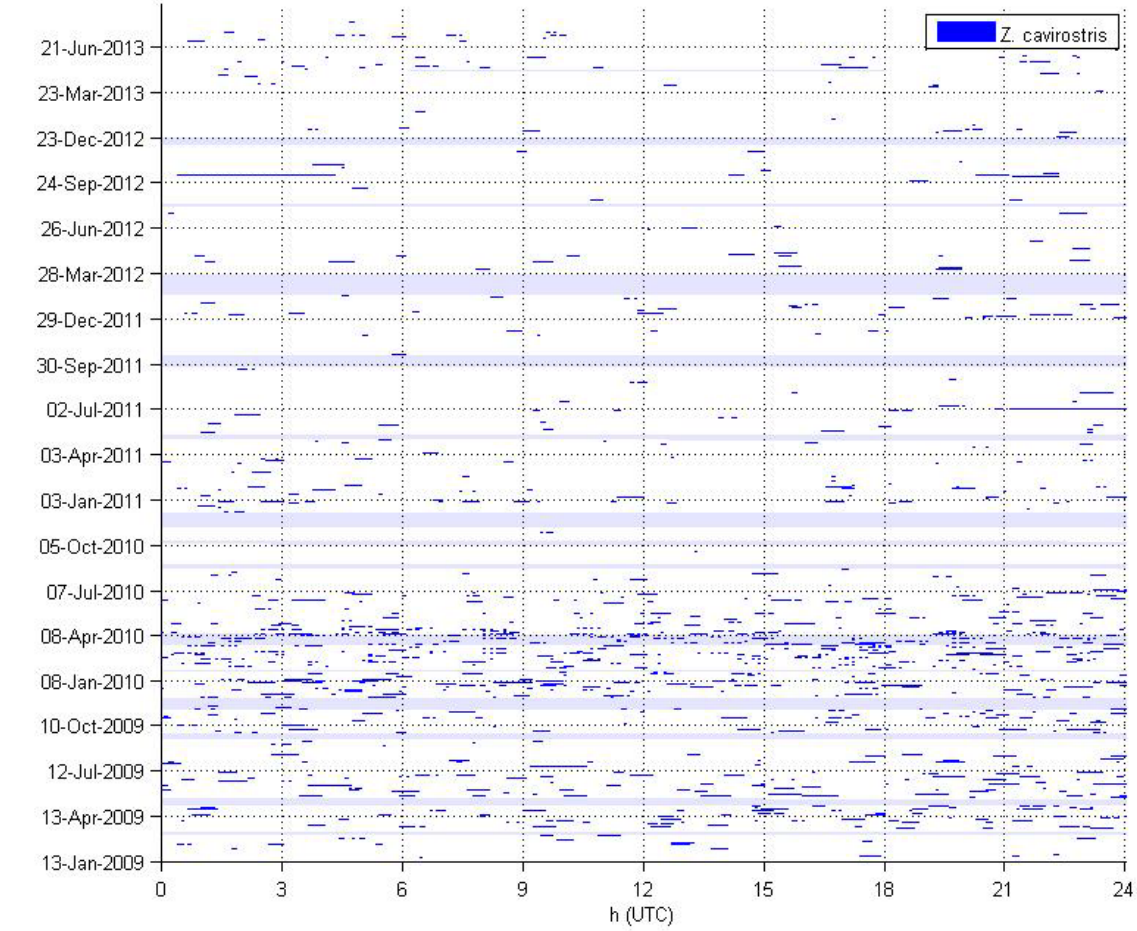
Returns

- det – Matrix of detection times
- endP – Sometimes detections don’t have an end time, this lets us know when do they do and when they don’t (one entry per detection)
- detinfo – Allows us to track detection times to specific deployments

Cuvier’s detections

h=visPresence(det, 'Resolution_m', 5, 'Effort', eff, 'DateTickInterval', 90)

legend(h(1), 'Z. cavirostris')



Data import

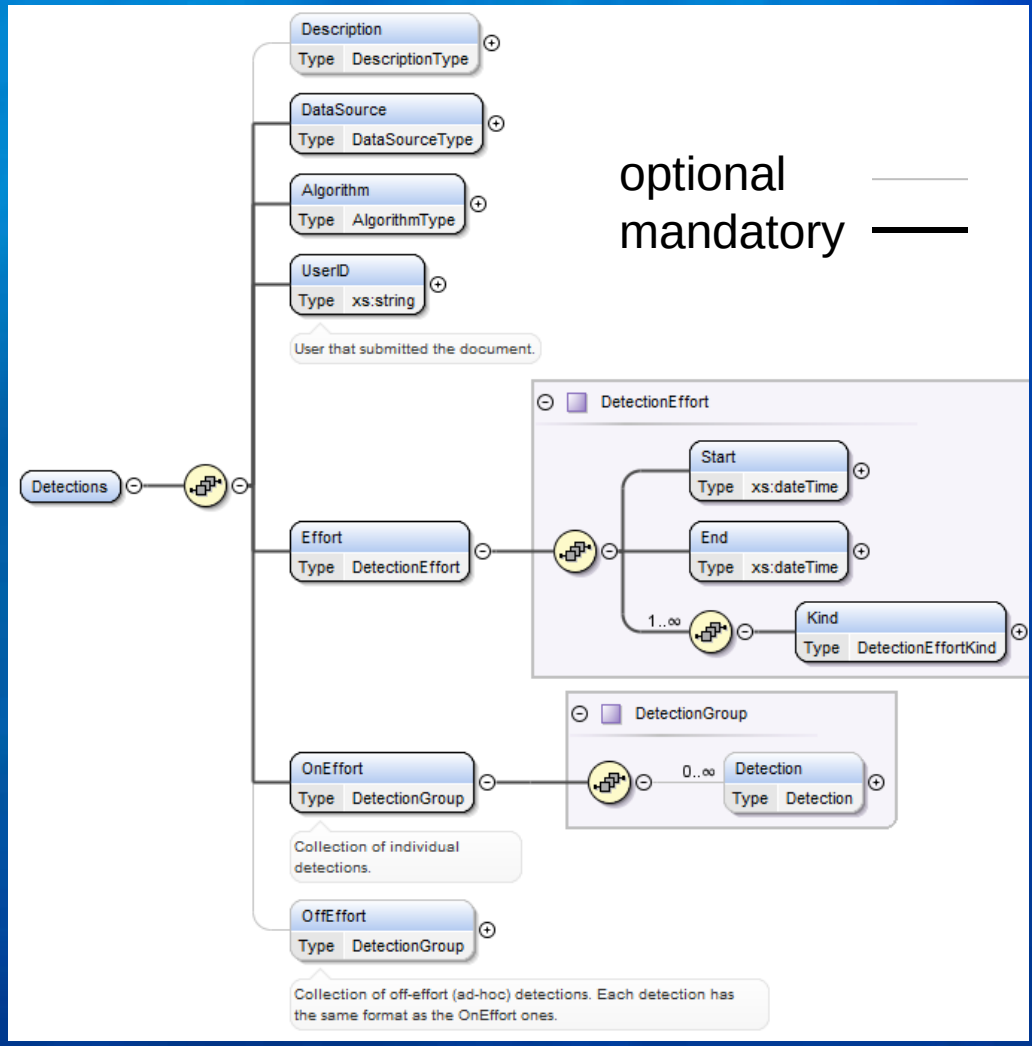
- Programmatically

  Java architecture for XML binding (JAXB)

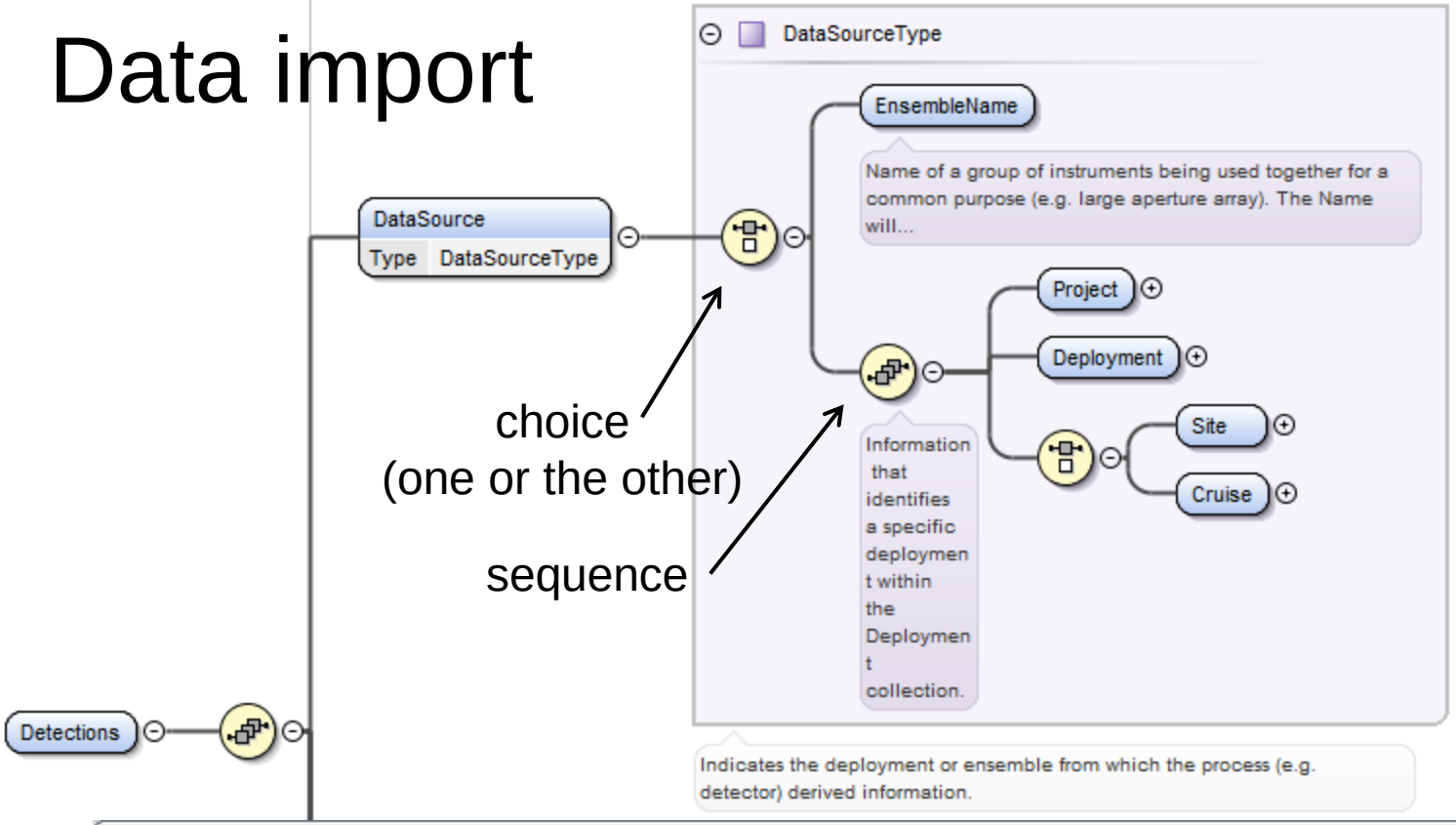
- Tabular format from a
  - database
  - comma separated value (CSV) file
  - spreadsheet

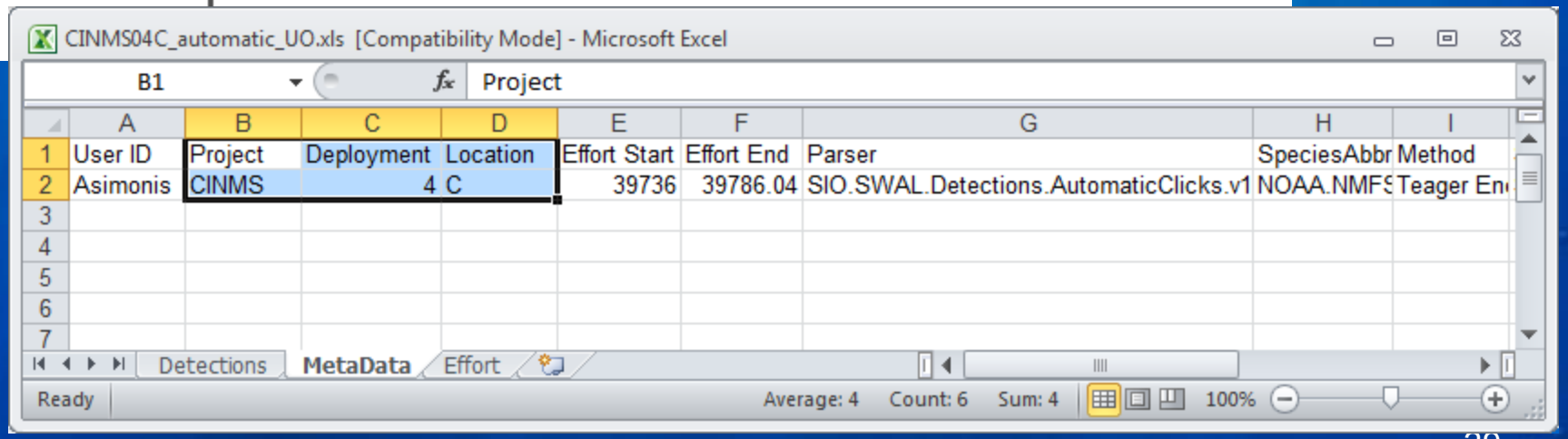
Data import

- Goal: Map our data onto the schema
- Example method:
  - Acoustic detections
  - spreadsheet source



Data import



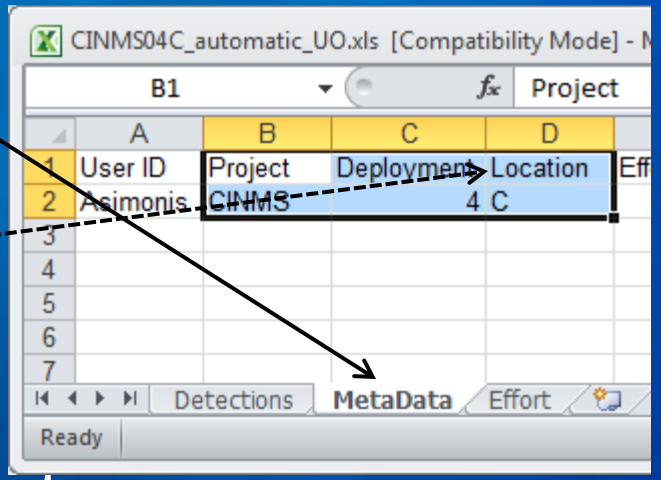


	A	B	C	D	E	F	G	H	I
1	User ID	Project	Deployment	Location	Effort Start	Effort End	Parser	SpeciesAbbr	Method
2	Asimonis	CINMS	4	C	39736	39786.04	SIO.SWAL.Detections.AutomaticClicks.v1	NOAA.NMFS	Teager En
3									
4									
5									
6									
7									

Data import source maps

- Match fields without programming:

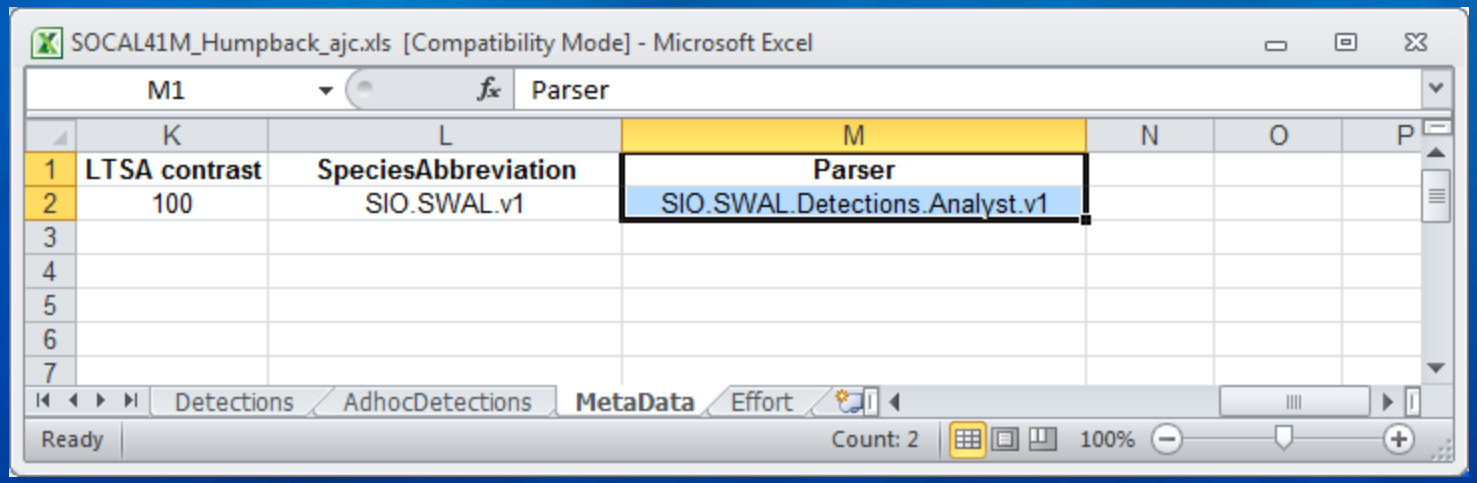
```
<Mapping>
 <Name>SIO.SWAL.Detections.Analyst.v1</Name>
 …
 <Directives>
    <Detections>  <!-- Name of document that we produce -->
      <Sheet name="MetaData">
         <Description> … </Description>
         <DataSource>
           <Entry> <Source> [Project] </Source> <Dest> Project </Dest> </Entry>
           <Entry>
             <Source> [Deployment] </Source>
             <Kind> integer </Kind>
             <Dest> Deployment </Dest>
           </Entry>
           <Entry>
             <Source> [Location] </Source>
             <Dest> Site </Dest>
           </Entry>
         </DataSource> …
```



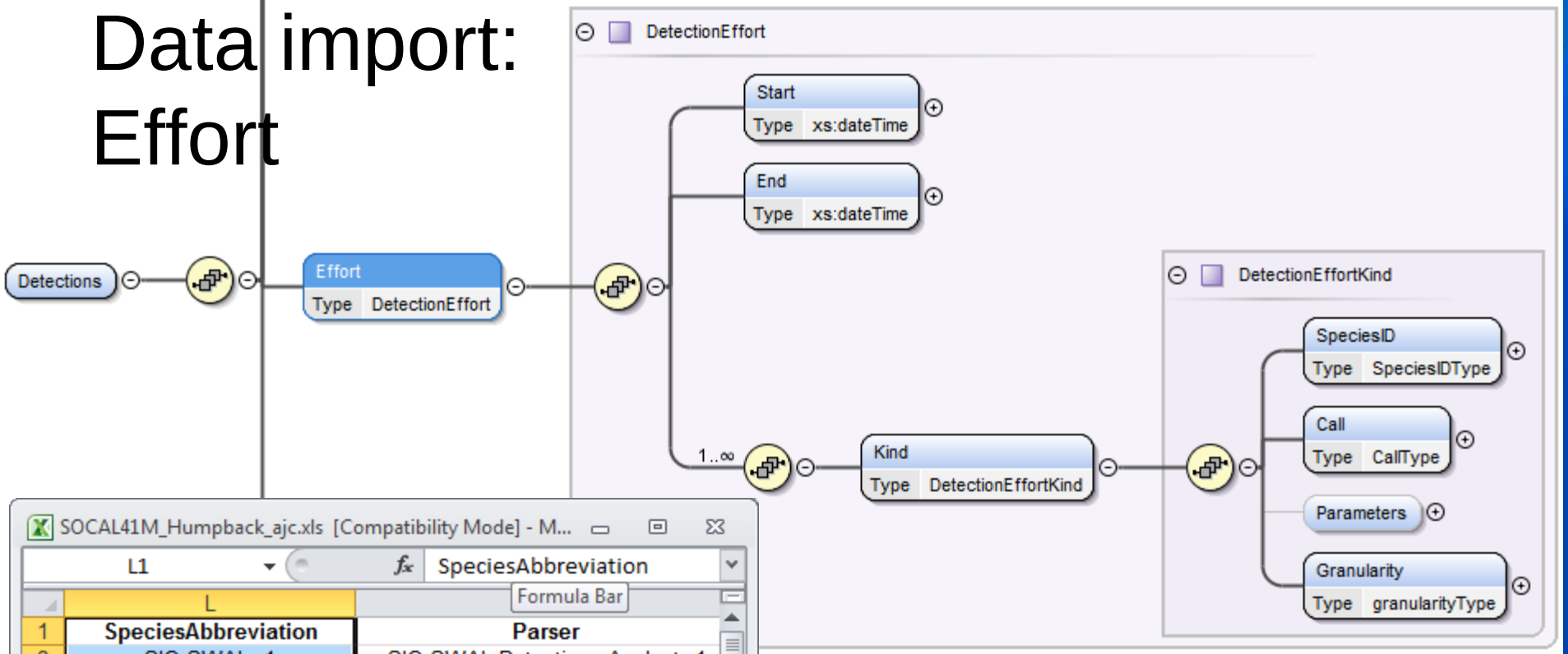
- Just another XML document

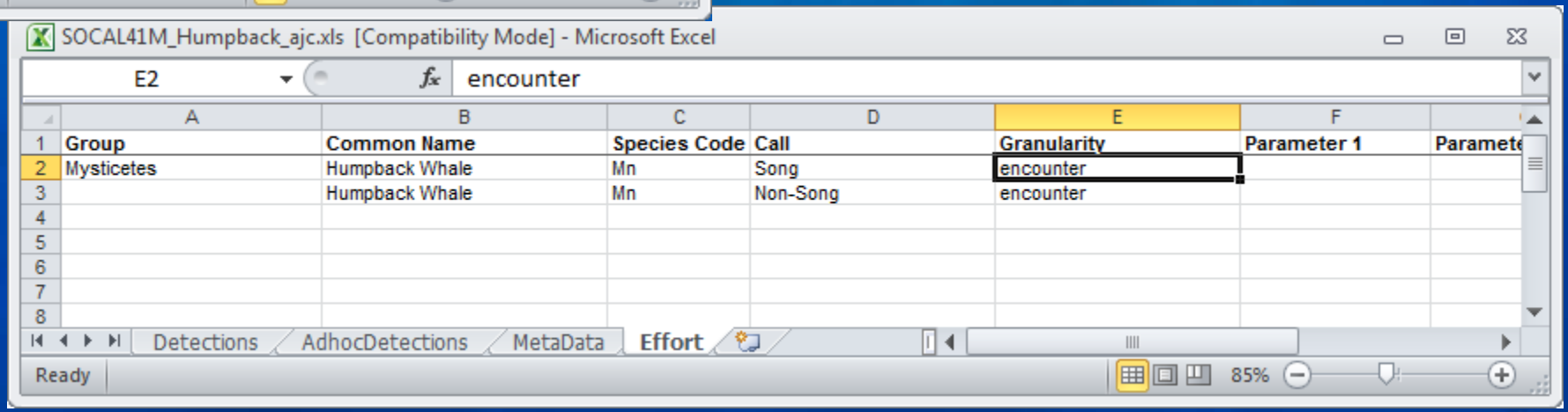
Data import

- These maps are inserted into the SourceMaps collection like any other document (see manual).
- A parser field in the spreadsheet MetaData tells Tethys which map to use.

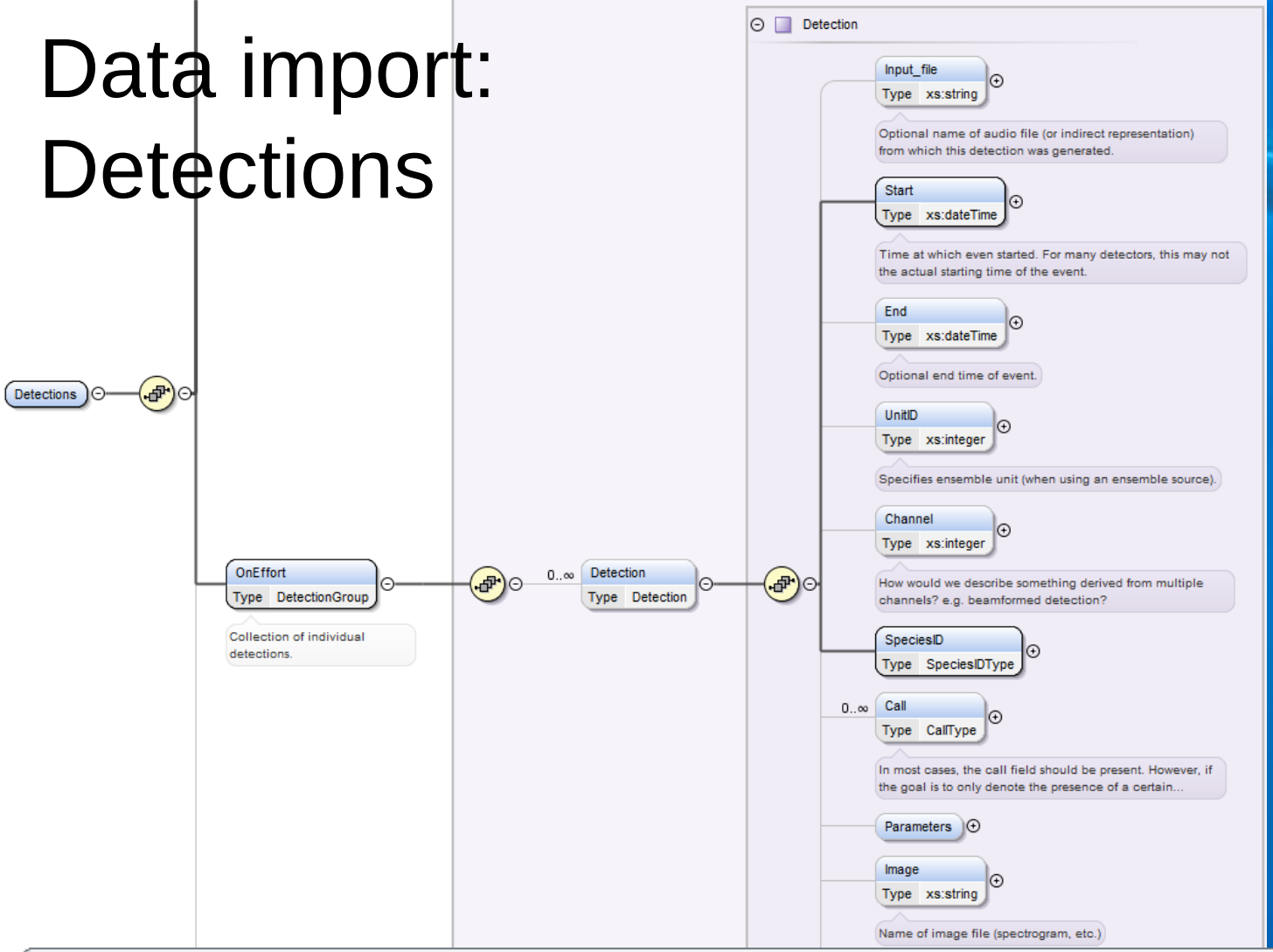


Data import: Effort





Data import: Detections



SOCAL41M_Humpback_ajc.xls [Compatibility Mode] - Microsoft Excel

A2 | SOCAL41M_

	A	B	C	D	E	F	G	H
1	Input file	Event Number	Species Code	Call	Start time	End time	Parameter 1	Parameter 2
2	SOCAL41M_	NED	Mn	Non-Song	12/15/2010 21:09:20	12/16/2010 2:37:25		
3	SOCAL41M_	NED	Mn	Non-Song	12/16/2010 7:16:00	12/16/2010 8:01:41		
4	SOCAL41M_	NED	Mn	Non-Song	12/16/2010 9:07:51	12/16/2010 9:07:52		

Detections | AdhocDetections | MetaData | Effort

Ready | Average: 40527.9954 | Count: 7 | Sum: 81055.9908 | 85%

Data import

Specifying the rows of data

- Readable by Excel:

  <Sheet name="sheet_tab">

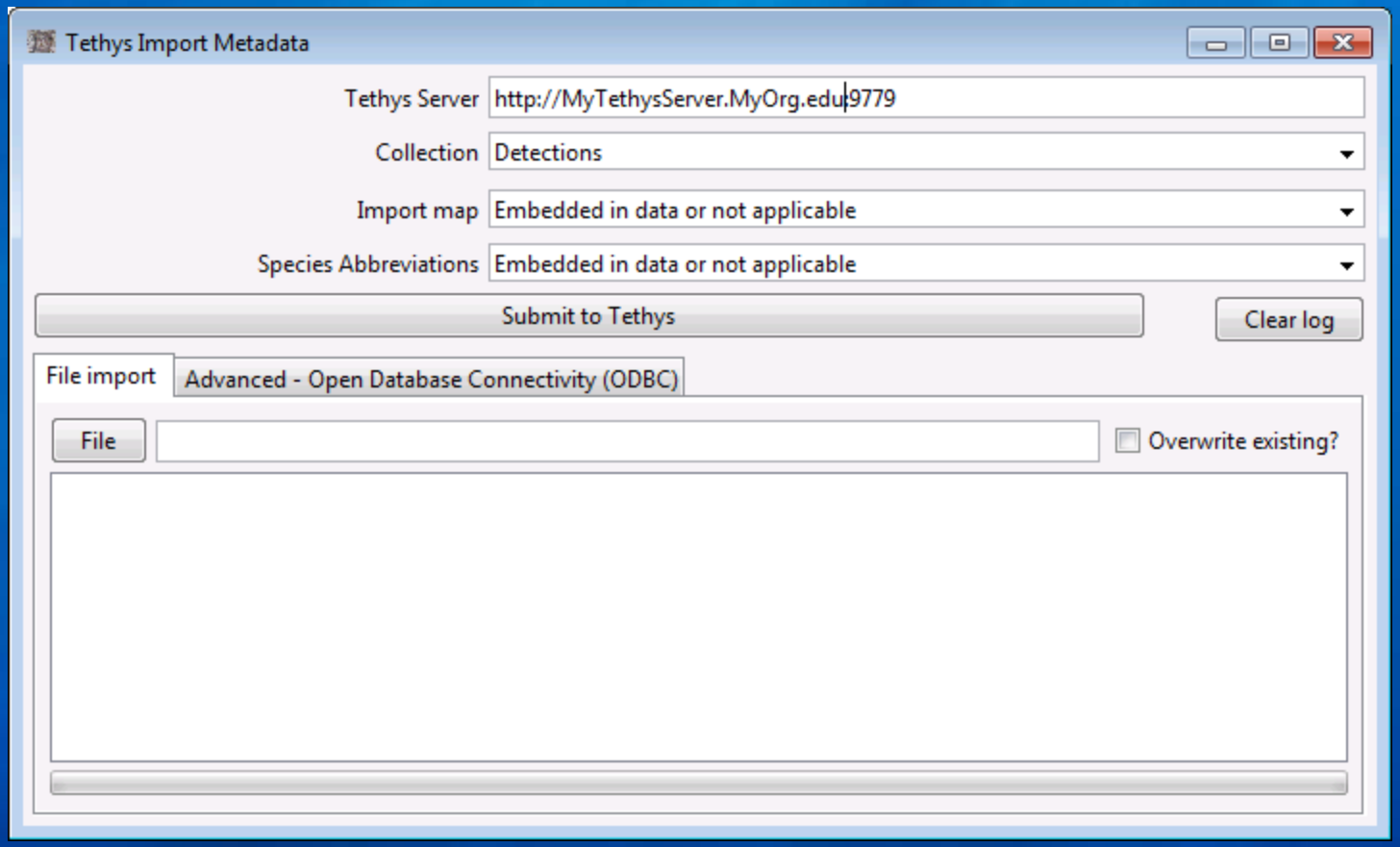
- Other types use open database connectivity (ODBC)

  <Table query="some SQL query">

ODBC is a technology to talk to databases . It typically requires separate software to be installed and specific parameters to initiate data transfer.

Data import

Java and Matlab graphical user interface



Looking at environmental data

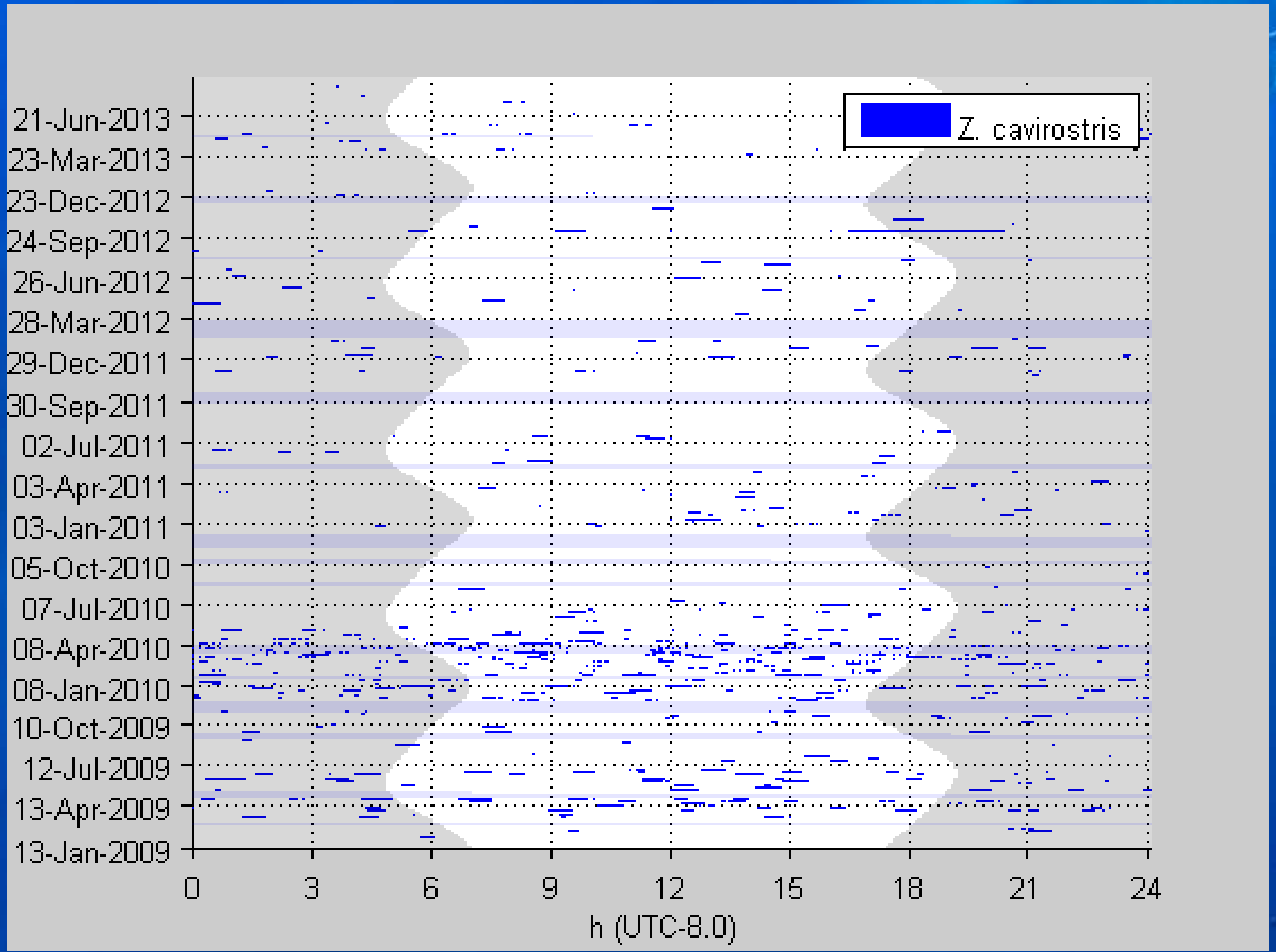
- Ephemeris – solar & lunar data
- SW NMFS ERDDAP – Wide range of data products
  - TAO buoys
  - NASA Ocean Color
  - CALCOFI
  - NODC
  - many others

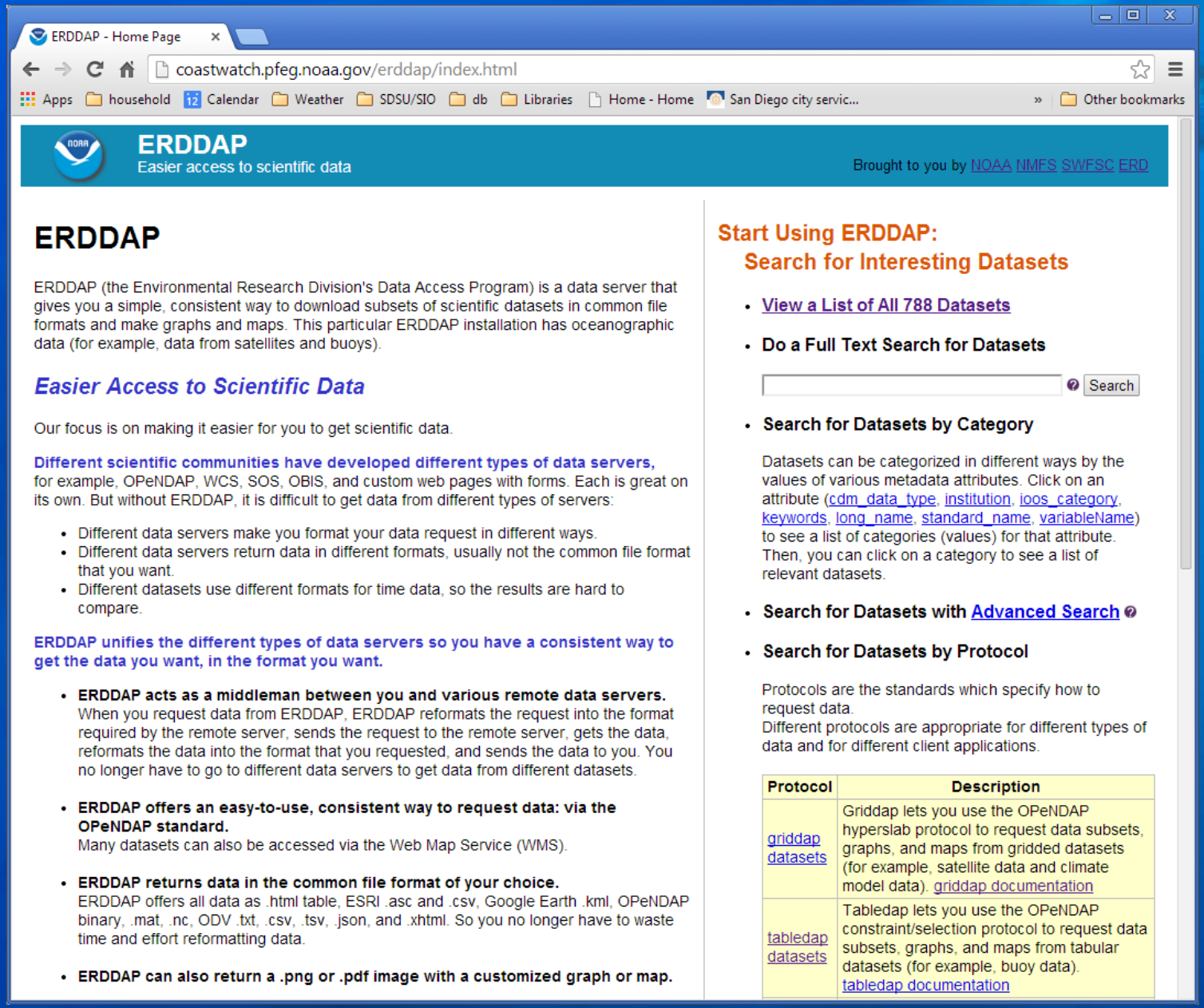
Ephemeris example

Site N: Long 241.435 E Lat 32.370 N

```
night = dbDiel(q, 32.370, 241.435,…
      min(eff(:,1)), max(eff(:,2)));
utcoffset = dbTimeZone(q, 32.370, 241.435);
nightH = visPresence(night, 'Color', …
  'black', 'UTCOffset', utcoffset, …
  'LineStyle', 'none', 'Transparency', …
  .15, 'Resolution_m', 1/60, 'DateRange', …
  EffortSpan);
h = visPresence(det, 'Resolution_m', 5, …
  'Effort', eff, 'DateTickInterval', 90, …
   'UTCOffset', utcoffset);
legend(h(1), 'Z. cavirostris')
```

local time


Z. cavirostris
21-Jun-2013
23-Mar-2013
23-Dec-2012
24-Sep-2012
26-Jun-2012
28-Mar-2012
29-Dec-2011
30-Sep-2011
02-Jul-2011
03-Apr-2011
03-Jan-2011
05-Oct-2010
07-Jul-2010
08-Apr-2010
08-Jan-2010
10-Oct-2009
12-Jul-2009
13-Apr-2009
13-Jan-2009
0
3
6
9
12
15
18
21
24
h (UTC-8.0)


ERDDAP - Home Page
coastwatch.pfeg.noaa.gov/erddap/index.html
Apps household Calendar Weather SDSU/SIO db Libraries Home - Home San Diego city servic... Other bookmarks
ERDDAP
Easier access to scientific data
Brought to you by NOAA NMFS SWFSC ERD
ERDDAP
ERDDAP (the Environmental Research Division's Data Access Program) is a data server that gives you a simple, consistent way to download subsets of scientific datasets in common file formats and make graphs and maps. This particular ERDDAP installation has oceanographic data (for example, data from satellites and buoys).
Easier Access to Scientific Data
Our focus is on making it easier for you to get scientific data.
Different scientific communities have developed different types of data servers, for example, OPeNDAP, WCS, SOS, OBIS, and custom web pages with forms. Each is great on its own. But without ERDDAP, it is difficult to get data from different types of servers:
Different data servers make you format your data request in different ways.
Different data servers return data in different formats, usually not the common file format that you want.
Different datasets use different formats for time data, so the results are hard to compare.
ERDDAP unifies the different types of data servers so you have a consistent way to get the data you want, in the format you want.
ERDDAP acts as a middleman between you and various remote data servers. When you request data from ERDDAP, ERDDAP reformats the request into the format required by the remote server, sends the request to the remote server, gets the data, reformats the data into the format that you requested, and sends the data to you. You no longer have to go to different data servers to get data from different datasets.
ERDDAP offers an easy-to-use, consistent way to request data: via the OPeNDAP standard.
Many datasets can also be accessed via the Web Map Service (WMS).
ERDDAP returns data in the common file format of your choice.
ERDDAP offers all data as .html table, ESRI .asc and .csv, Google Earth .kml, OPeNDAP binary, .mat, .nc, ODV .txt, .csv, .tsv, .json, and .xhtml. So you no longer have to waste time and effort reformatting data.
ERDDAP can also return a .png or .pdf image with a customized graph or map.
Start Using ERDDAP:
Search for Interesting Datasets
View a List of All 788 Datasets
Do a Full Text Search for Datasets
Search
Search for Datasets by Category
Datasets can be categorized in different ways by the values of various metadata attributes. Click on an attribute (cdm_data_type, institution, ioos_category, keywords, long_name, standard_name, variableName) to see a list of categories (values) for that attribute. Then, you can click on a category to see a list of relevant datasets.
Search for Datasets with Advanced Search
Search for Datasets by Protocol
Protocols are the standards which specify how to request data.
Different protocols are appropriate for different types of data and for different client applications.
Protocol
Description
griddap datasets
Griddap lets you use the OPeNDAP hyperslab protocol to request data subsets, graphs, and maps from gridded datasets (for example, satellite data and climate model data). griddap documentation
tabledap datasets
Tabledap lets you use the OPeNDAP constraint/selection protocol to request data subsets, graphs, and maps from tabular datasets (for example, buoy data). tabledap documentation

ERDDAP

1. Identify what type of data
   - Search the ERDDAP site or
   - Use Matlab dbERDDAPSearch.
2. Read about the data set
   - Understand limitations
   - Recognize there may be gaps
   - Verify temporal-spatial coverage and resolution are appropriate.

ERDDAP Requests

- Select variables to return, e.g. SST
- Select temporal-spatial ranges by:
  - index
  - value (in parenthesis)
- Ranges have a stride, or increment between points:
  - 20:2:40 – Indices 20, 22, 24, 26,… along axis
  - (2013-12-28T12:00:00Z):1:(2013-12-31T12:00:00Z)

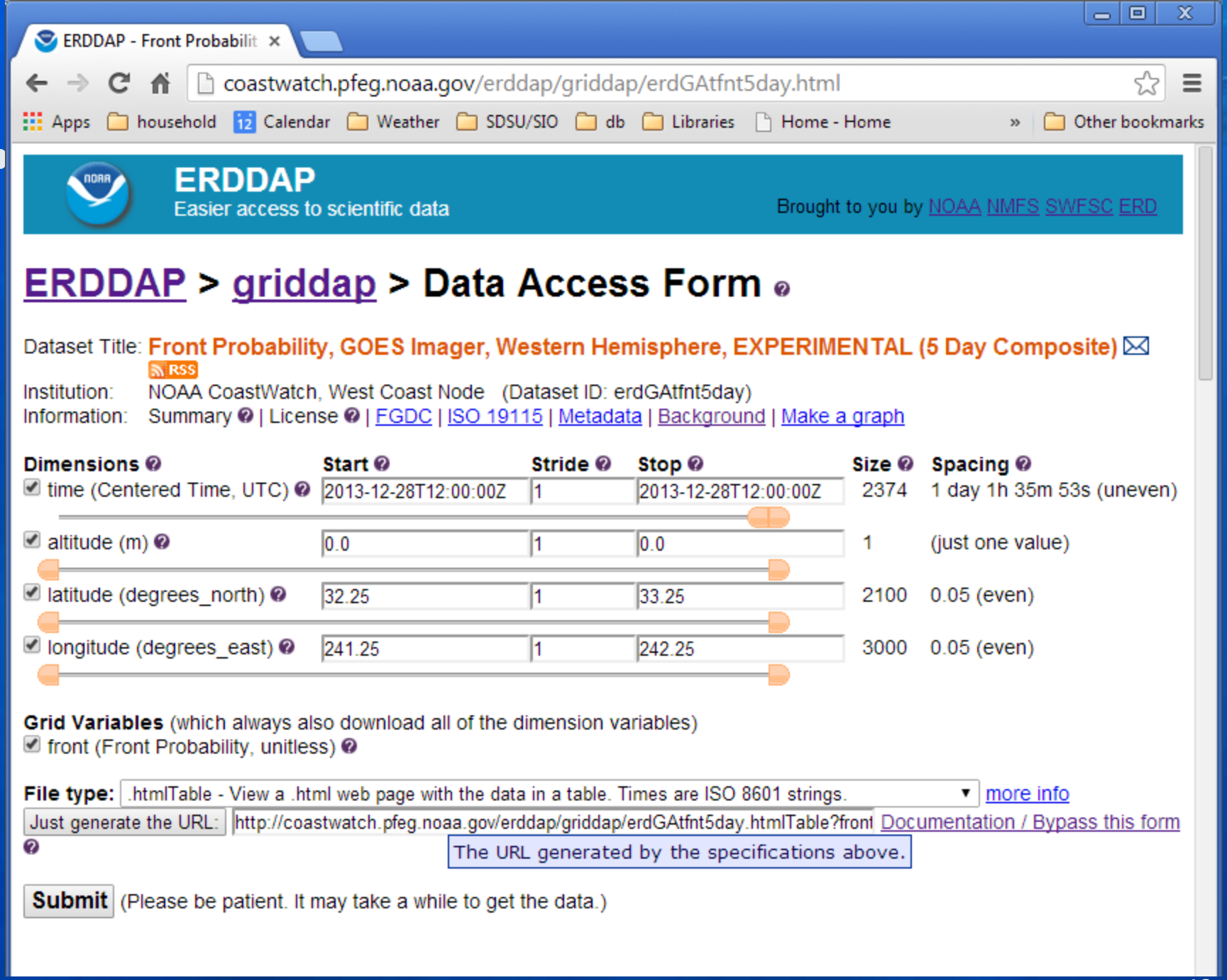
ERDDAP example

Suppose we are interested in weather fronts:

```
dbERDDAPSearch(q, 'keywords=front')
```

We see that there is a 5 day average front probability from a GOES satellite with dataset id: `erdGAtfnt5day`

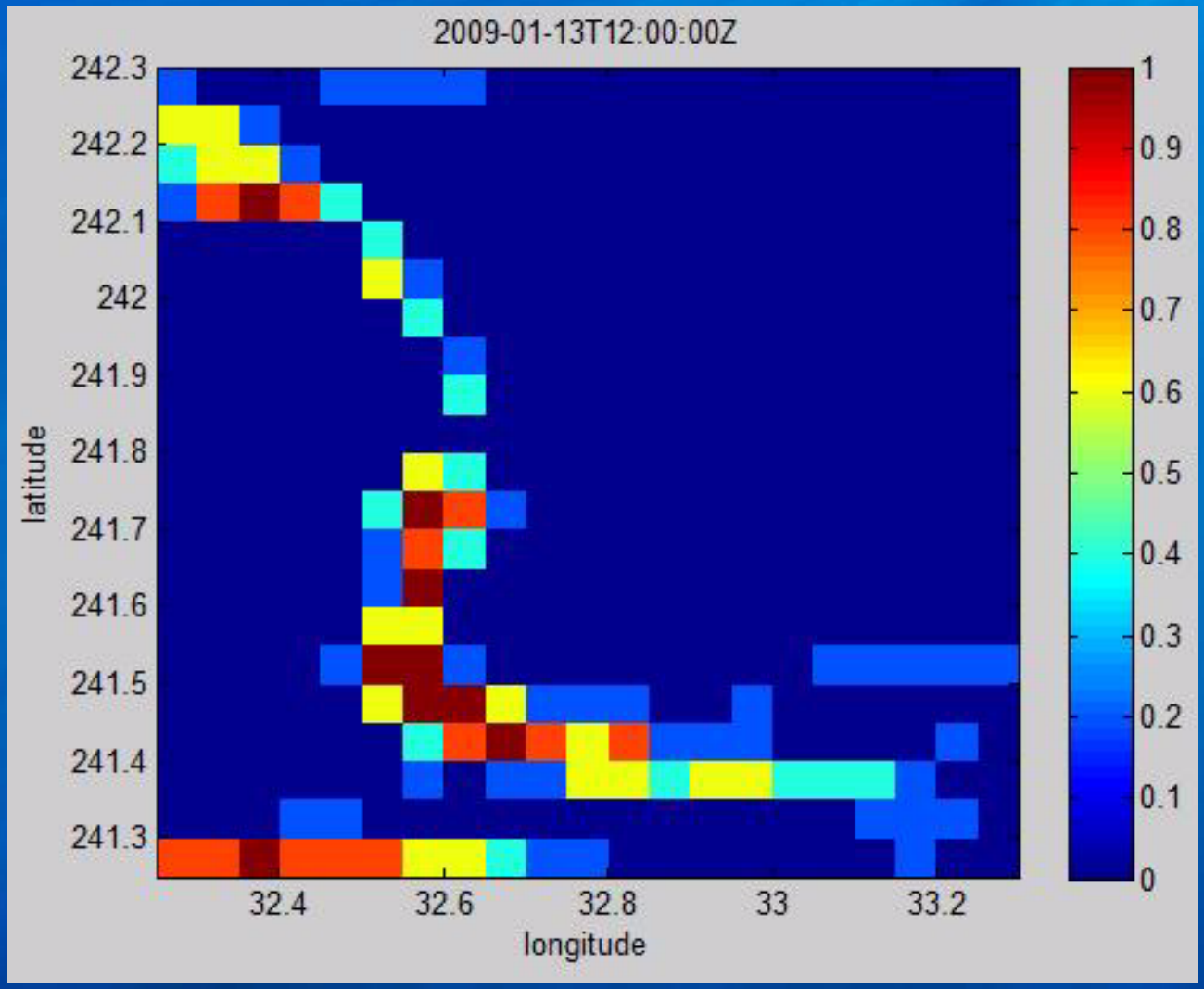
We'll use the ERDDAP web site to get an idea of how to query it.


ERDDAP - Front Probabilit
coastwatch.pfeg.noaa.gov/erddap/griddap/erdGAtfnt5day.html
Apps household Calendar Weather SDSU/SIO db Libraries Home - Home Other bookmarks
ERDDAP
Easier access to scientific data
Brought to you by NOAA NMFS SWFSC ERD
ERDDAP > griddap > Data Access Form
Dataset Title: Front Probability, GOES Imager, Western Hemisphere, EXPERIMENTAL (5 Day Composite)
RSS
Institution: NOAA CoastWatch, West Coast Node (Dataset ID: erdGAtfnt5day)
Information: Summary | License | FGDC | ISO 19115 | Metadata | Background | Make a graph
Dimensions Start Stride Stop Size Spacing
time (Centered Time, UTC) 2013-12-28T12:00:00Z 1 2013-12-28T12:00:00Z 2374 1 day 1h 35m 53s (uneven)
altitude (m) 0.0 1 0.0 1 (just one value)
latitude (degrees_north) 32.25 1 33.25 2100 0.05 (even)
longitude (degrees_east) 241.25 1 242.25 3000 0.05 (even)
Grid Variables (which always also download all of the dimension variables)
front (Front Probability, unitless)
File type: .htmlTable - View a .html web page with the data in a table. Times are ISO 8601 strings. more info
Just generate the URL: http://coastwatch.pfeg.noaa.gov/erddap/griddap/erdGAtfnt5day.htmlTable?front Documentation / Bypass this form
The URL generated by the specifications above.
Submit (Please be patient. It may take a while to get the data.)

Weather fronts around San Clemente Island

```
Pfront = dbERDDAP(q, …
     sprintf('erdGAtfnt5day?front[(%s):1:(%s
)][(0.0):1:(0.0)][(32.25):1:(33.25)][(241.25
):1:(242.25)]', …
   dbSerialDateToISO8601(EffortSpan(1)), …
   dbSerialDateToISO8601(EffortSpan(2))))
Pfront =
    Axes: [1x1 struct]
    Data: [1x1 struct]
    dims: [21 21 1 1644]
```

Weather fronts around San Clemente Island



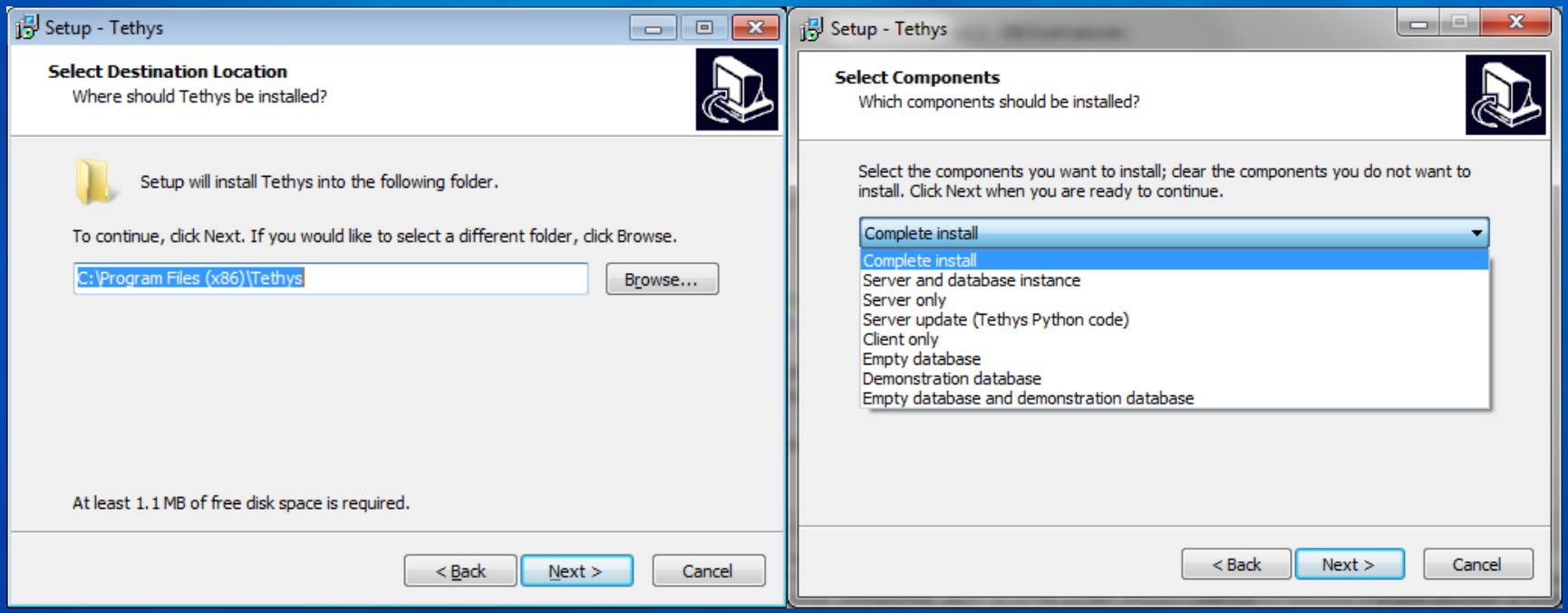
How we generated the movie

```
h = figure();
for d=1:size(Pfront.Data.values{1}, 4)
    % Data comes back as long/lat/altitude/time
    % We want latitude going up/down, so we will transpose the long/lat
    % matrix.  For each plot, altitude and time are constant, so we
    % "squeeze" out the singleton dimensions in the 4 D data.
    imagesc(Pfront.Axes.values{2}, Pfront.Axes.values{1}, ...
        squeeze(Pfront.Data.values{1}(:, :, 1, d)'), [0 1]);
    set(gca, 'YDir', 'normal');  % imagesc plots upside down, fix
    colorbar % Show probability scale
    xlabel(Pfront.Axes.names{1});
    ylabel(Pfront.Axes.names{2});
    title(dbSerialDateToISO8601(Pfront.Axes.values{4}(d)));
    drawnow  % force image to update
    frames(d) = getframe(h);  % grab a frame to save a movie file
end
m = VideoWriter('Pfront.avi') % write the movie file
open(m); for d=1:length(frames); writeVideo(m, frames(d)); end; close(m);
```

Tethys installation

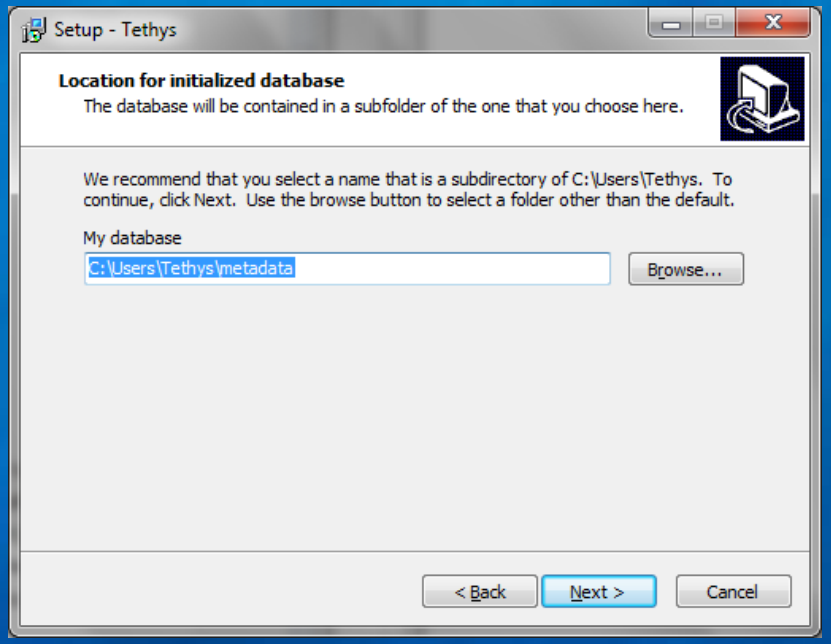
Highlights – Details in the Tethys manual

Download: http://tethys.sdsu.edu



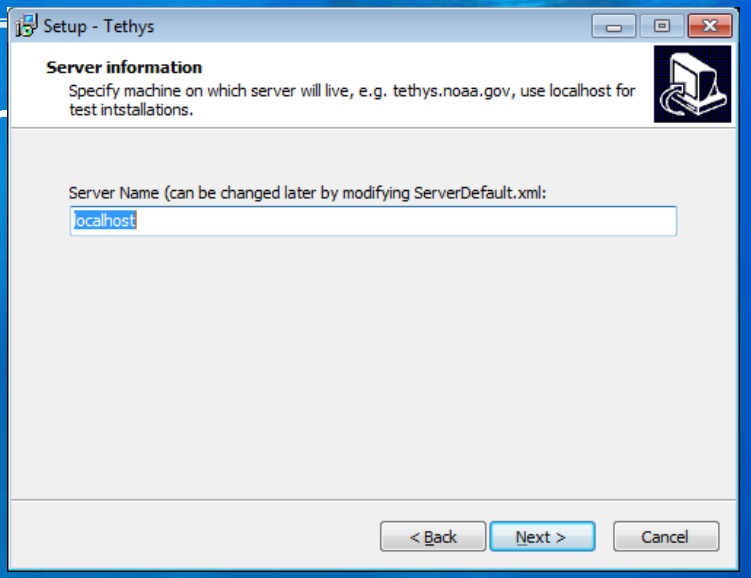
Tethys installation

If databases are requested, data locations must be selected:

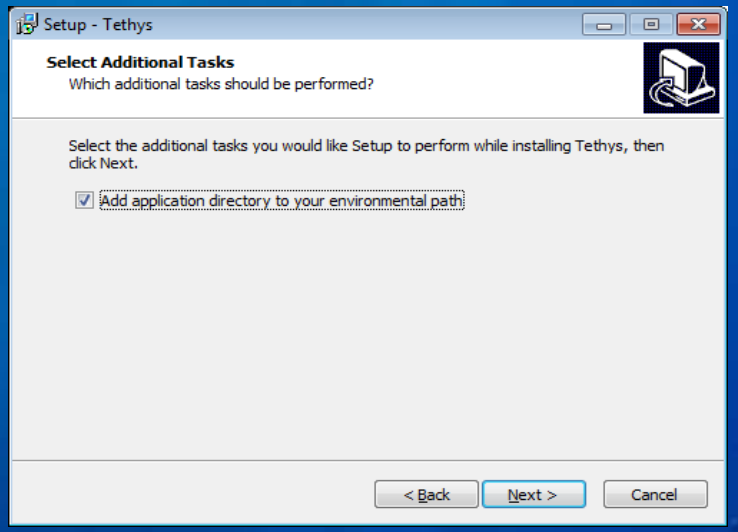


Tethys installation

- When asked for a server name, either allow localhost or specify the server's name, e.g.: bowhead.afsc.noaa.gov



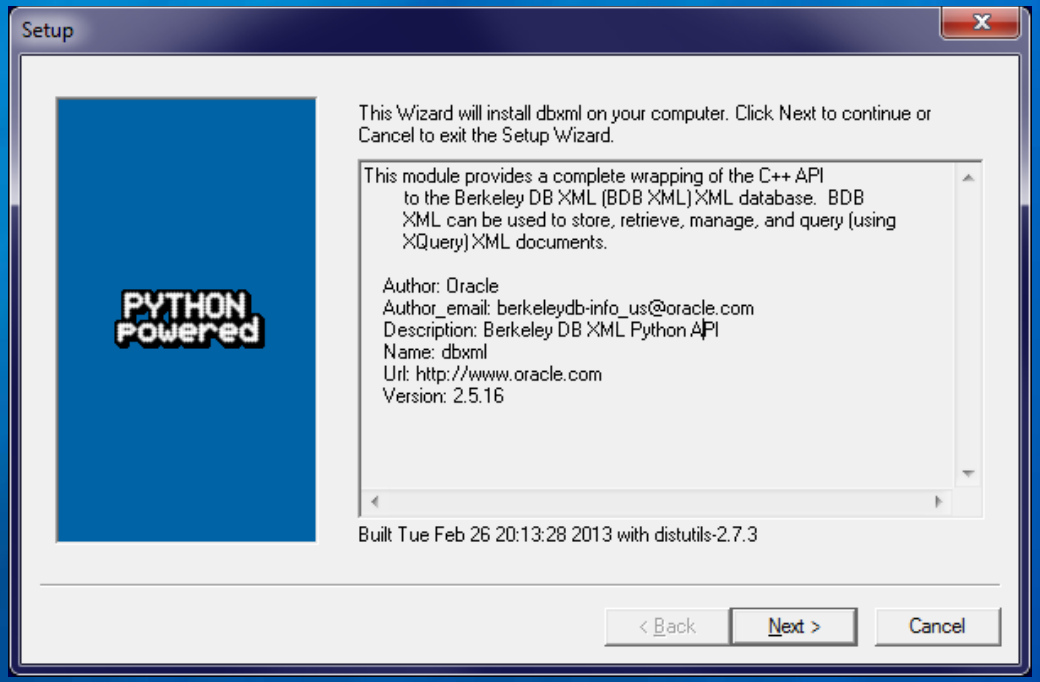
- You will be asked if you wish to add Python to your path, say yes:



Installing Tethys

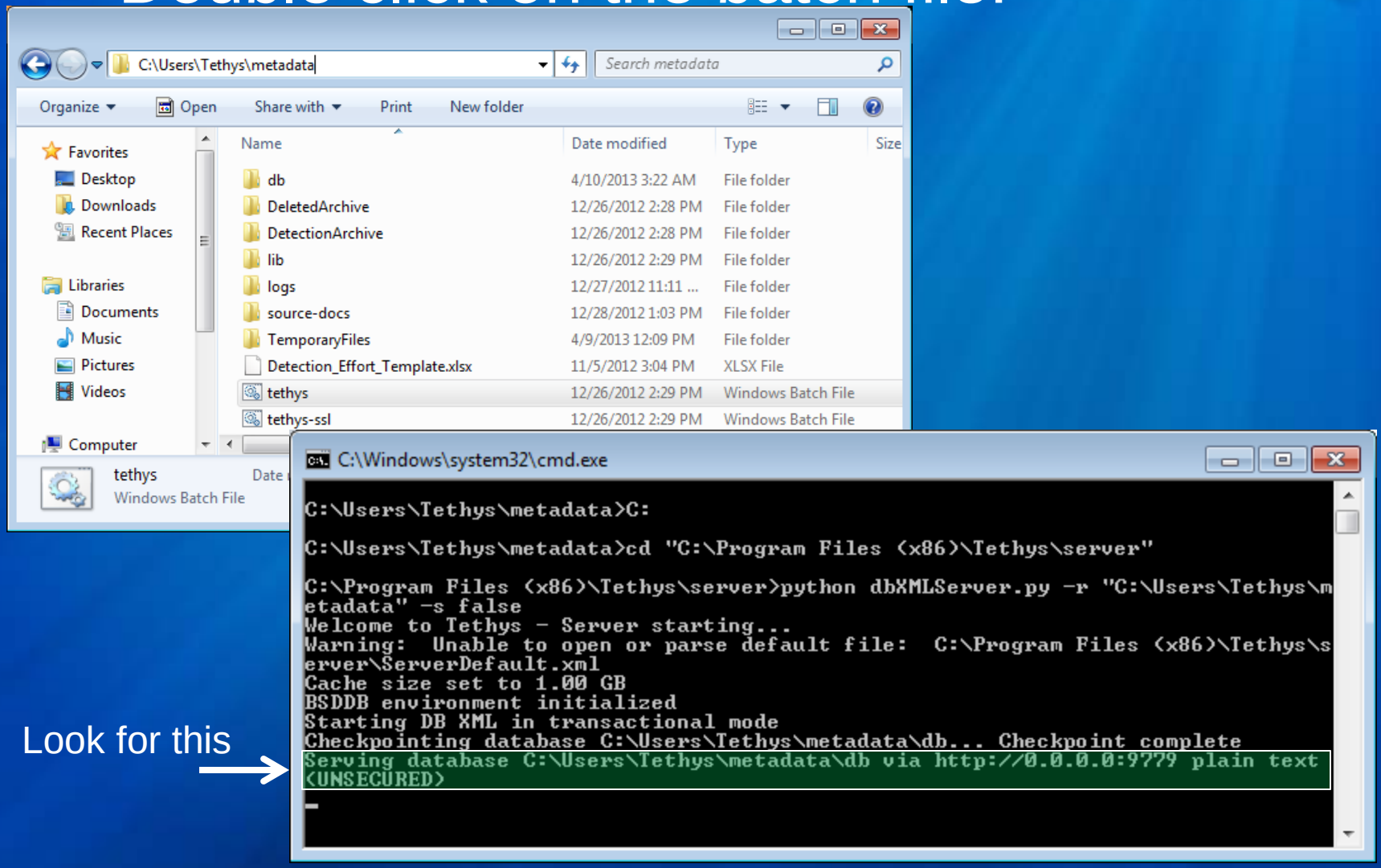
Several other installers will launch and *must* complete successfully, install to default locations:

- Python
- Egenix
- PyWin32
- PyODBC
- PyBSDDB
- PyDBXML



Starting Tethys

- Double click on the batch file:



Look for this →

Didn’t start?

- Window gone → no server

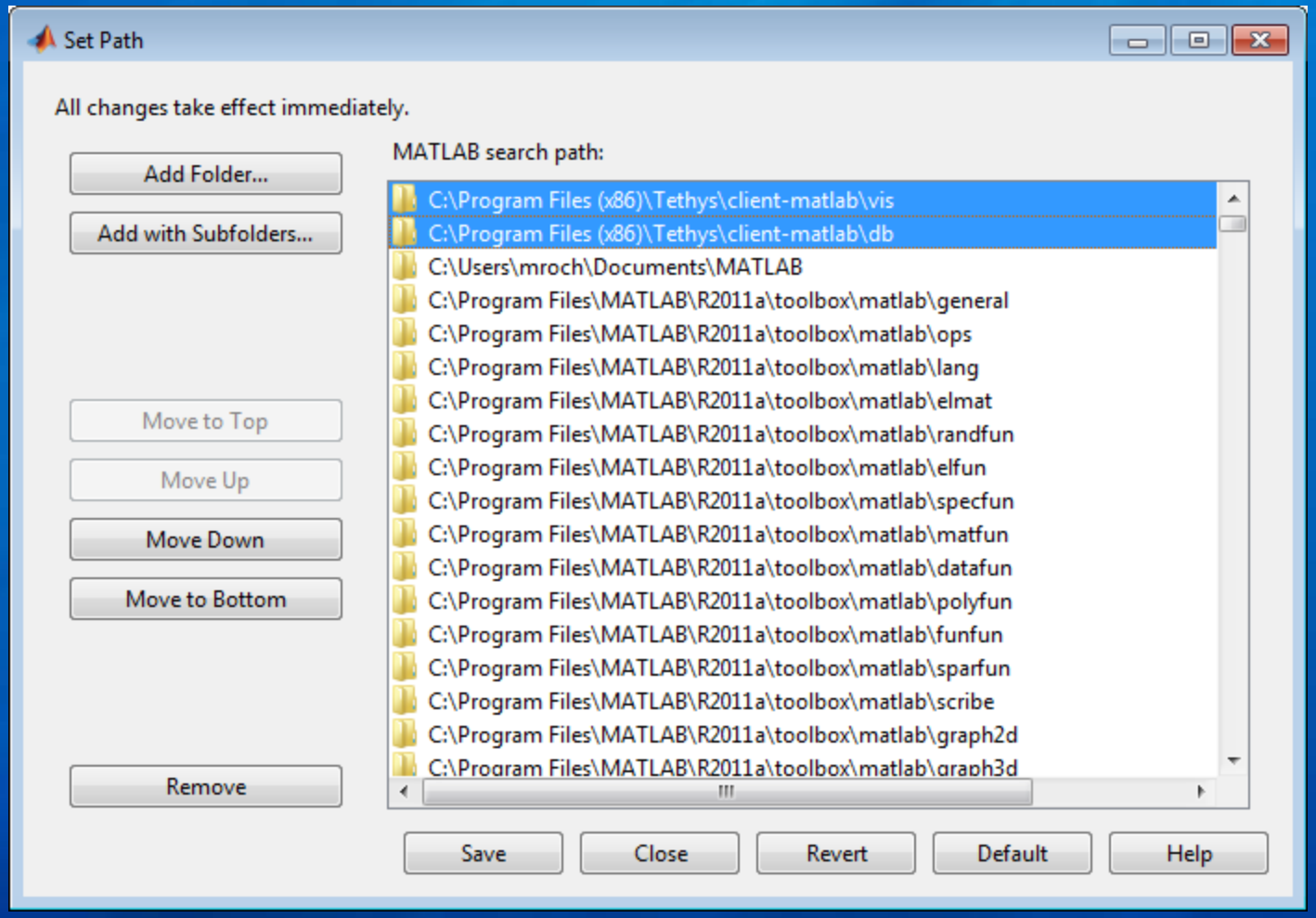
- To determine the problem, open a command window then:

```
cd c:\Users\Tethys\metadata
tethys.bat
```

Note: The path may differ if you installed the database to a different folder.

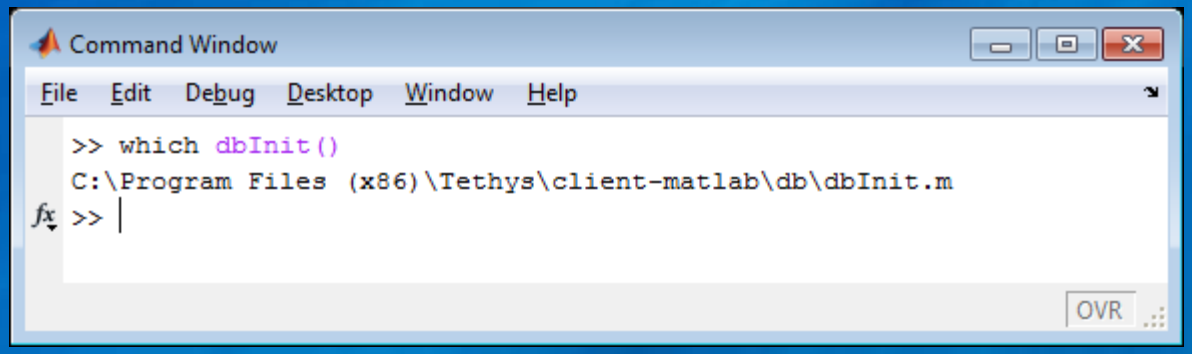
Setting up Matlab

Make sure the Matlab files are on your path



May be different if 64 bit version installed or you overrode default directories

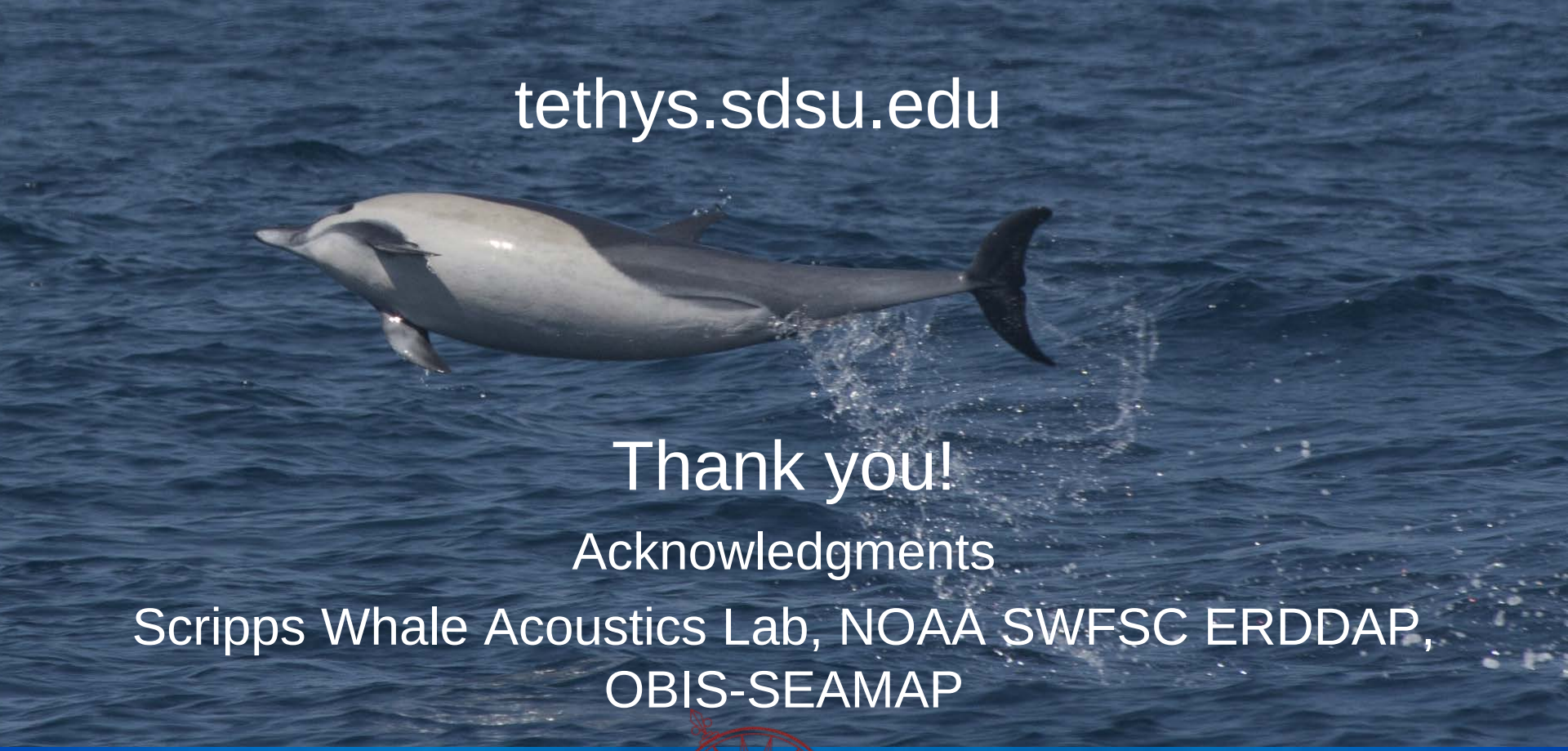
Verifying Matlab paths



tethys.sdsu.edu

What's in the download:

- Small sample database
- Server
- Clients
- Manual
- Cookbook



tethys.sdsu.edu

Thank you!

Acknowledgments

Scripps Whale Acoustics Lab, NOAA SWFSC ERDDAP, OBIS-SEAMAP

Michael Weise & Dana Belden

National Oceanographic Partnership Program

Jill Lewandowski and Jim Price

data – NAVFAC Living Marine Resources
Applied Research Bob Gisiner & Frank Stone