The Use of Web Development Technology as a Method for Visual Analysis of Seaglider Acoustic and Environmental Data

By

Amanda Walsh

15 December 2022

University of Rhode Island

Abstract

This research project explores the use of web development technology as an interface for analyzing passive acoustic SONAR audio and environmental data captured from a Seaglider Autonomous Underwater Vehicle. User-friendly, responsive, and flexible web applications with interactive graphical data visualizations can be used as an open-source alternative to other data visualization applications, like MATLAB. A proof-of-concept “Glider Data Viewer” web application was developed to analyze the glider’s passive acoustic audio recordings, as well as the glider’s timestamped depth, temperature, salinity, speed, latitude, and longitude measurements. The resulting application demonstrates that the JavaScript Web Audio API (application programming interface) is a suitable method for creating spectrogram data from acoustic audio files using the built in Fast Fourier Transform (FFT) method. The LightningChart JavaScript library was used to create fast, responsive, and easy-to-use displays for visualizing the glider’s data and spectrograms. This paper explains the Web Audio API signal processing functions, the libraries used for data visualization, and the methods used for developing the web application. Examples from the Gulf of Mexico 2019 dataset are displayed to demonstrate the utility of the application, as well as a User Guide for the Glider Data Viewer application.

Table of Contents

Table of Figures

Introduction

The Glider Data Viewer allows for the import, processing, and display of NetCDF files and audio files in an interactive, user-friendly web application. The primary use case for this application is for the display and analysis of data collected from seaglider dives, including position, environmental, and passive acoustic data. A web application is a program that runs in a web browser and is accessed online or through a locally hosted server. The Glider Data Viewer App can be accessed online at https://awalsh64.github.io/glider/ where it is hosted with GitHub Pages.

Signal Processing

Audio Processing Capabilities

The Glider Data Viewer allows the user to import one or more passive acoustic audio files, select FFT parameters to process the files, and display the resulting spectrograms. Accepted formats for the audio files include wav (Waveform Audio File Format) or mp3. In order for the app to set a timestamp for the audio file, the file name must follow the timestamp naming convention of “yymmdd_hhmmss.wav” (or .mp3) where the date and time specifies the start time of the audio file, and the hours are measured out of 24. The required file name values can be preceded with any number of characters.

The user can define the parameters for performing the Fast Fourier Transform (FFT) on the audio files to adjust the frequency, time, and amplitude ranges and resolutions. The NFFT parameter sets the number of frequency points used to calculate the FFT. Increasing this parameter will increase the resolution in the frequency domain. The Processor Buffer Size parameter specifies the length of data chunks provided to the frequency analyzer. Decreasing this parameter will increase the resolution in the time domain [2]. The NFFT and Processor Buffer Size values can be adjusted to balance the resolution and the processing time and power required by the computer. They must be a power of two between 256 and 16384. The Sample Rate parameter is defined in hertz and can be in the range of 3000 to 768000 Hz. It can match the sample rate of the audio file, or down sample to decrease processing time, but will cause a failure if set greater than the audio file sampling rate. The sample rate controls the maximum frequency of the spectrogram, which is equal to half the sampling rate. The Minimum dB and Maximum dB parameters specify the scaling range for the amplitude of the output data. The Maximum dB value must be less than or equal to 0 and greater than -100dB. The Minimum dB value must be less than the Maximum dB value. The amplitude colormap for the spectrogram will be adjusted to match the decibel range.

Processing and displaying the audio files with high resolution is a memory intensive process. The length of audio and resolution to which it can be processed is dependent on the available memory of the computer used. The FFT parameters should be adjusted to produce lower resolution spectrograms if many files are loaded.

The NFFT is defaulted to 2048, and the Sample Rate is defaulted to 48000, and the Processor Buffer Size is defaulted to 1024. This produces a frequency resolution of approximately 24Hz, a time resolution of approximately 0.022 seconds, and a maximum frequency of 24kHz. The Maximum dB value is defaulted to -60dB and the Minimum dB is defaulted to -160dB.

Web Audio API Processing Functions

The Web Audio API (application programming interface) provides a powerful interface for controlling audio on the Web [11]. This API can be used to play audio, modify the audio with effects, and analyze the audio to create visualizations. Functions are available in this JavaScript API for reading in audio files and converting them to the frequency domain.

To use the Web Audio API for audio file processing, an AudioContext is created which contains the audio processing, or decoding, functions [11]. The wav (Waveform Audio File Format) or .mp3 audio file is loaded into the browser from an HTML input attribute using an XML HttpRequest of type ArrayBuffer. An ArrayBuffer is a JavaScript array of bytes representing raw binary data that cannot be directly manipulated. The Web AudioContext is used to decode the audio data ArrayBuffer at the specified sample rate [11].

An OfflineAudioContext is created to process the audio data as fast as possible, in contrast to the normal AudioContext which would process the data at the playback speed [11]. The OfflineAudioContext is created with the predefined sample rate, and the source is assigned to the audio file's ArrayBuffer decoded data.

An AnalyserNode is created from the OfflineAudioContext, and the audio file's source is connected to the analyzer to generate the Fast Fourier Transform data. The AnalyserNode parameters are the FFT (Fast Fourier Transform) size, a smoothing time constant which smooths the value changes over time (set to 0 so no smoothing is performed), and a minimum and maximum decibel value which specifies the scaling range for the FFT analysis data.

The AnalyserNode uses the getByteFrequencyData function to get the frequency data of the decoded audio file data chunks. The Processor Buffer Size parameter controls how frequently the chunks of audio data are processed [9]. The process that the Web Audio API uses to get the frequency data is detailed by W3C [10]. First, the time-domain data is computed for the audio file. A Blackman window is applied to the time domain input data. A Fourier transform is applied to the windowed time domain data, producing real and imaginary frequency data. The frequency domain data is smoothed if a smoothing time constant is specified. The frequency data is converted to decibels and clipped between the minimum and maximum specified decibel values. It is then scaled to fit in an unsigned byte for storage by converting it to an integer from 0 to 255 representing the decibel value for each frequency, from 0 to half the sampling rate. [10]

The Analyser's frequency bin count is equal to half the FFT size, which controls the frequency resolution of the spectrogram. If the frequency bin count is less than the number of frequencies in the range, the frequency resolution is scaled by the number of frequency bins [10]. For example, a sample rate of 128kHz will allow for a frequency range of 64kHz. A FFT size of 4096 will allow for a frequency bin count of 2048. The 2048 bins will be scaled to cover the frequency range, producing a frequency resolution between 31 and 32Hz on the spectrogram for this example.

Other methods of performing the FFT on the audio files were investigated to introduce more flexibility for parameters like windowing and overlapping. Because the Web Audio API's getByteFrequencyData function performs all the steps internally as detailed previously, it does not allow these parameters to be adjusted. Other node packages for performing the FFT were tested, but none demonstrated better performance than the Web Audio API for working with large audio files in the browser. Future development of the Web Audio API may introduce customizable windowing functions, as suggested in the open Web Audio API issue 2421 that can be tracked on GitHub, https://github.com/WebAudio/web-audio-api/issues/2421.

Environmental Data Processing

Glider NetCDF Files

Seagliders measure environmental parameters throughout operation, and record timestamped data variables in Network Common Data Form (NetCDF) files. Seaglider depth, latitude, and longitude and water temperature and salinity are measured. The Glider Data Viewer imports NetCDF v3.x files and plots these values in various displays.

The NetCDF variable names used for plotting are "ctd_time", "ctd_depth", "temperature", "salinity_raw", "latitude", and "longitude". The temperature, salinity_raw, and ctd_depth measurements were used to calculate the sound speed using the Mackenzie Equation [7],

$$c(D,S,T) = \\ 1448.96 + 4.591T - 5.304\,x\,10^{-2}T^2 + 2.374\,x\,10^{-4}T^3 + 1.340\,(S-35) + \\ 1.630\,x\,10^{-2}D + 1.675\,x\,10^{-7}D^2 - 1.025\,x\,10^{-2}T(S-35) - 7.139\,x\,10^{-13}TD^3,$$

where T is temperature in degrees Celsius, S is salinity in parts per thousand, and D is depth in meters. The range of validity is 2 to 30 °C for temperature, 25 to 40 parts per thousand for salinity, and 0 to 8000 meters for depth. If these variables are not present in the file, an error will be thrown.

The NetCDF.js package version 2.0.1 was utilized to read and convert the NetCDF variables into JavaScript for processing in the app [8]. This package is compatible with NetCDF v3.x files.

Bathymetric NetCDF File

Bathymetry data can be imported into the Glider Data Viewer in the form of one NetCDF file with variables for latitude, longitude, and elevation or depth. The default import settings for the app are defined for the 2D NetCDF Grid from the General Bathymetric Chart of the Oceans (GEBCO) Gridded Bathymetry Data Download website, https://download.gebco.net/.

If other sources for NetCDF bathymetry files are used, the variable names can be specified, and the direction of positive depth/elevation measurements. Generally, elevation measurements are positive above sea level. These measurements are converted to be positive depth measurements below sea level for display in the app. The format of the data is expected to be an array of latitude values and an array of longitude values covered by the gridded bathymetry, and an array of depth or elevation values at each corresponding longitude and latitude point in order of rows. For example,

Longitude = [lon1, lon2, …]

Latitude = [lat1, lat2, …]

Depth = [depth-lat1-lon1, depth-lat1-lon2, depth-lat1-lon3, …].

Data Visualization

Environmental Data Plots

The Geo Location plot displays the latitude and longitude of the glider's path over time in the global space. The x-axis is longitude, and the y-axis is latitude. The path is colormapped for the glider's depth. If bathymetry is imported, the Geo plot will display a heatmap of the bathymetry under the glider's path. Hovering over the glider path will provide a tooltip for the latitude (degrees), longitude (degrees), glider depth (meters), and glider speed (centimeters per second). Hovering over the bathymetric heatmap will provide a tooltip for the latitude (degrees), longitude (degrees), and ocean depth (meters).

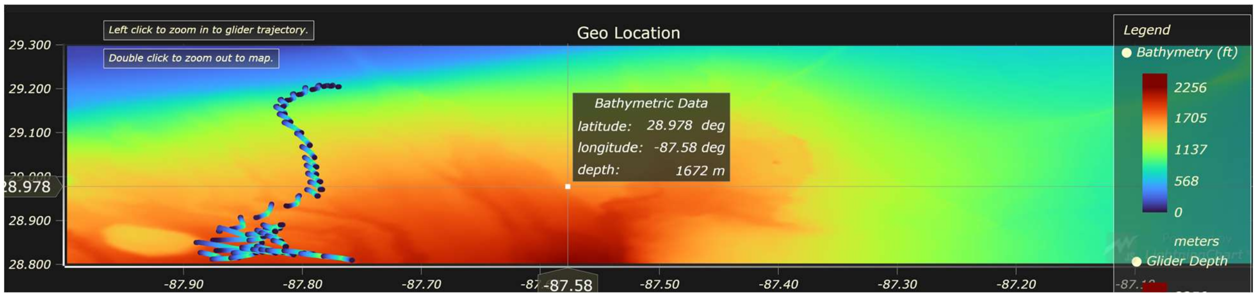


Figure 1: Geo Location plot with Bathymetric Data tooltip on hover.

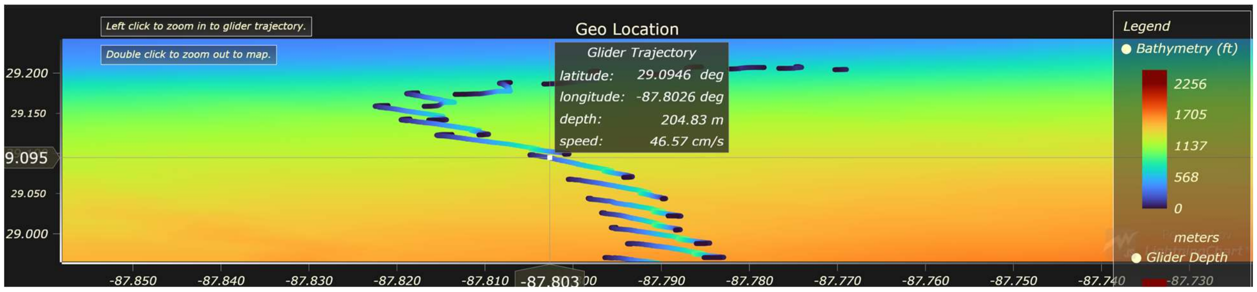


Figure 2: Geo Location plot with Glider Trajectory tooltip on hover.

The Glider Trajectory plot displays the glider's depth over time with a colormap for the calculated sound speed in meters per second at each timestamp. The x-axis is GMT time and the y-axis is depth in meters. The Temperature and Salinity plot displays the temperature and salinity at each timestamp measured by the glider. The x-axis is GMT time, the left y-axis is temperature in degrees Celsius, and the right y-axis is salinity in parts per thousand. Tooltips are shown when hovering over both the Glider Trajectory and the Temperature and Salinity plots that provide the GMT time, glider depth (meters), temperature (degrees Celsius), salinity (parts per thousand), and sound speed (meters per second).

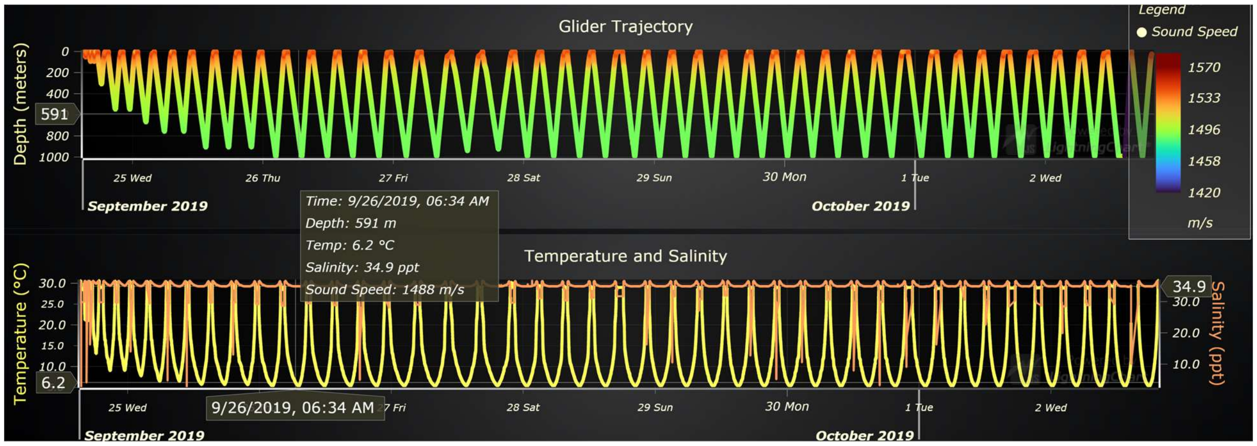


Figure 3: Glider Trajectory and Temperature and Salinity plots with hover tooltips.

Spectrogram Display

The Spectrogram plot displays a heatmap with time on the x-axis in the format of hours, minutes, and seconds (hh:mm:ss), frequency on the y-axis in hertz, and a colormap for the acoustic data Power/Frequency (dB/Hz). Tooltips are displayed when hovering over the Spectrogram plot that provide the time (hh:mm:ss), frequency (hertz), and amplitude (dB).

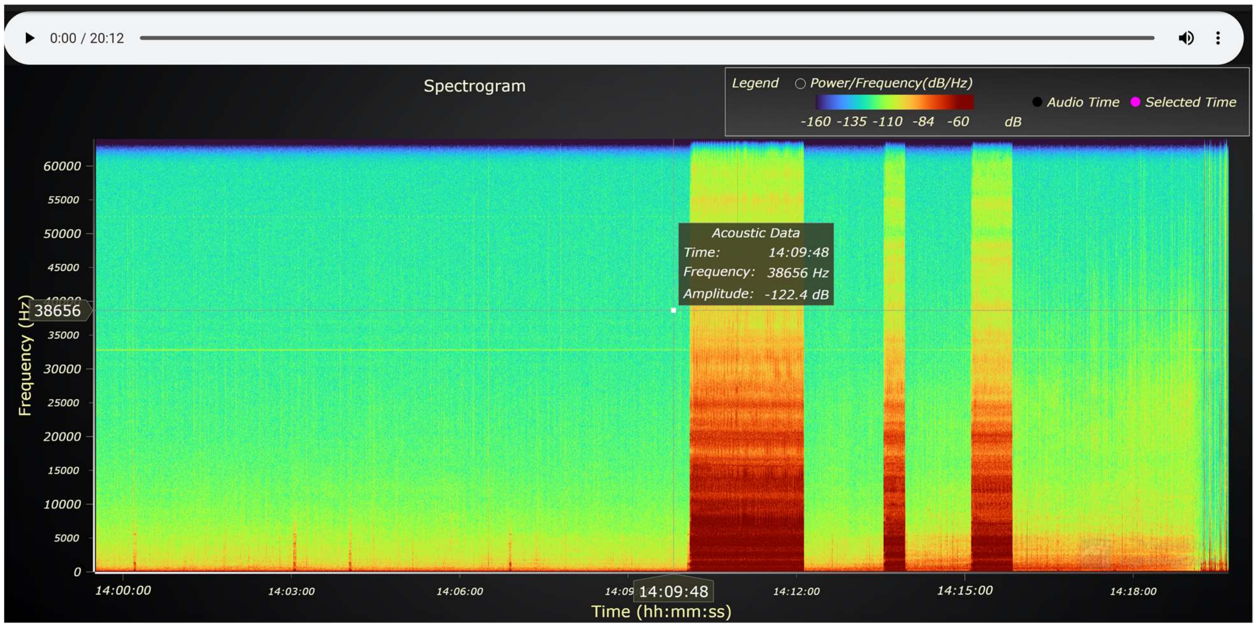


Figure 4: Spectrogram Display with hover tooltip.

The audio for each passive acoustic recording can be played for the current selected audio file. While playing the audio, a black line on the Spectrogram plot will be drawn to signify the current audio time. Clicking on the Spectrogram plot will adjust the audio playback time to the selected time. Depending on the resolution of the spectrogram, the audio playback time marker may only be accurate within a few hundred milliseconds due to the resolution of the HTML audio player.

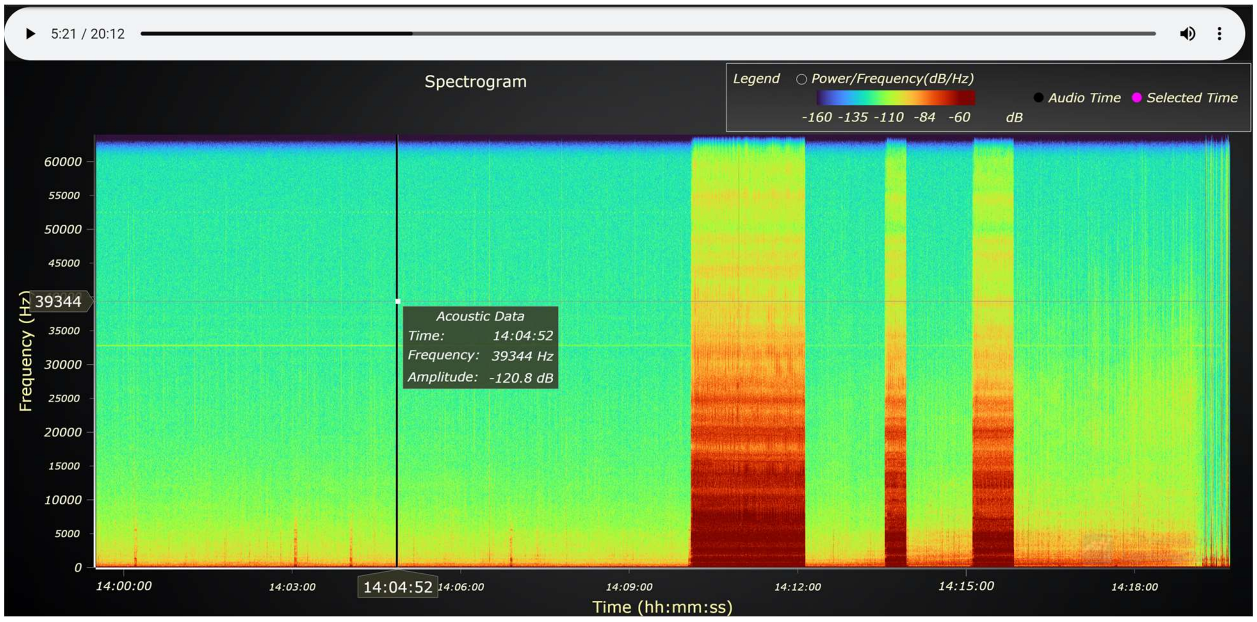


Figure 5: Spectrogram Display with Playback Time Line.

The displayed audio file can be selected from the Audio Files list or by clicking the Glider Trajectory or Temperature and Salinity plots. When both NetCDF and audio files are loaded, the start time of each the spectrogram will be displayed on the environmental plots with a cyan line. Clicking the plots will load and display the spectrogram that starts prior to the selected time, and the selected time will be marked with a magenta line. If the selected time is within the duration of the spectrogram, a magenta line will also be displayed on the spectrogram.

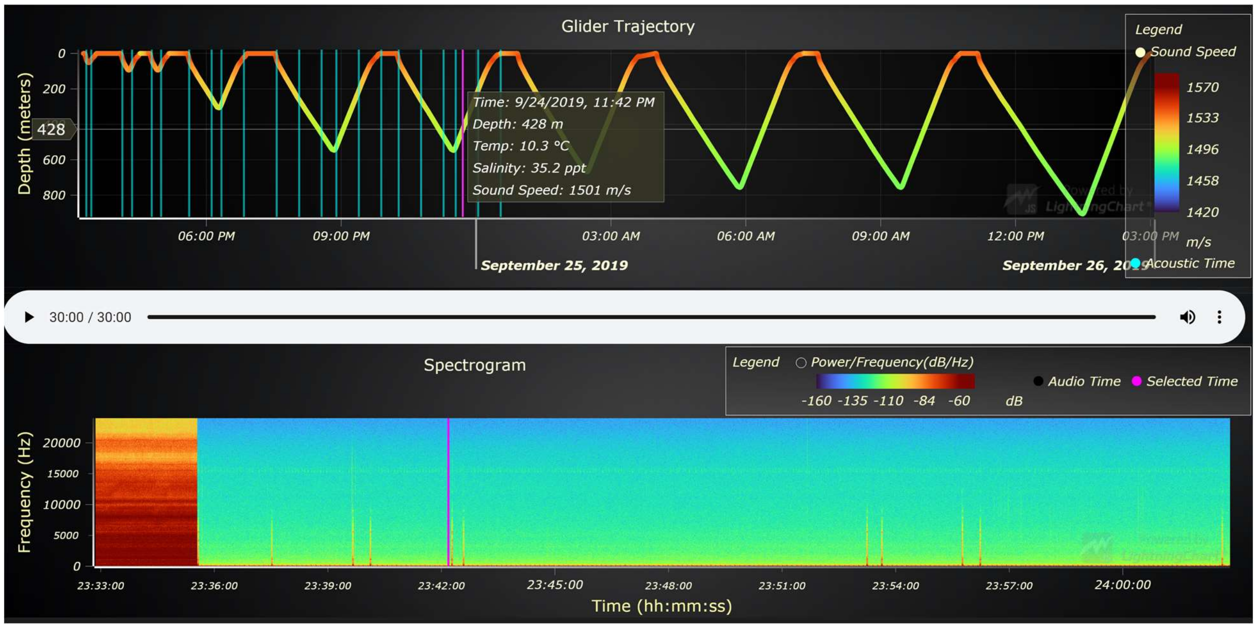


Figure 6: Click Glider Trajectory or Temperature and Salinity Plots to select Spectrogram.

The Lightningchart.js example for "JavaScript Heatmap Spectrogram Chart" was used as a starting point for the spectrogram chart [6]. The example was modified to work in the Vue.js framework using the lcjs-vue-template repository on GitHub as a guide. The best practices for this setup include generating a random chart ID to use for the chart container ID and target div ID before the component is mounted, creating the chart after the component is mounted, and disposing the chart when the component is unmounted [1].

LightningChart JavaScript Plotting Library

The "lightning-fast", interactive, and responsive JavaScript plotting library, LightningChart JS, was implemented for development of the visual displays. LightningChart JS utilizes GPU acceleration and WebGL rendering to maximize performance for millions to billions of data points, making it a leader in engineering and scientific data visualization [5].

The LightningChart JS ChartXY and MapChart were used for the various displays. The ChartXY was used with the LineSeries for line plots like the Glider Trajectory and Temperature and Salinity, and the HeatmapGridSeries for the spectrogram and bathymetry heatmap plots. The MapChart was used with the full world image overlaid with a ChartXY to plot the location of the glider in latitude and longitude coordinates. Documentation for modifying the parameters of each plot can be found for the version 3.4.0 used at https://lightningchart.com/lightningchart-js-api-documentation/v3.4.0/index.html.

LightningChart is used with the free Community License, which is suitable for universities, not commercial use. This version includes the product watermark in the bottom right corner of the plots [4].

Observations in Gulf of Mexico 2019 Data

Figure 7 shows a measurement error that occurred on October 2nd from 10:35am to 11:49am between dive 45 and 46. Figure 7 uses the “ctd_depth” variable to plot the glider’s depth and highlights the time of the error with a red box. The glider uncharacteristically remained at max depth instead of rising to the surface, then depth data was unavailable from 10:35am to 11:49am. The temperature measurement was also undefined just before the missing data, causing the line to be colormapped purple for the undefined sound speed value. The measurements recover after this time and continues as expected.

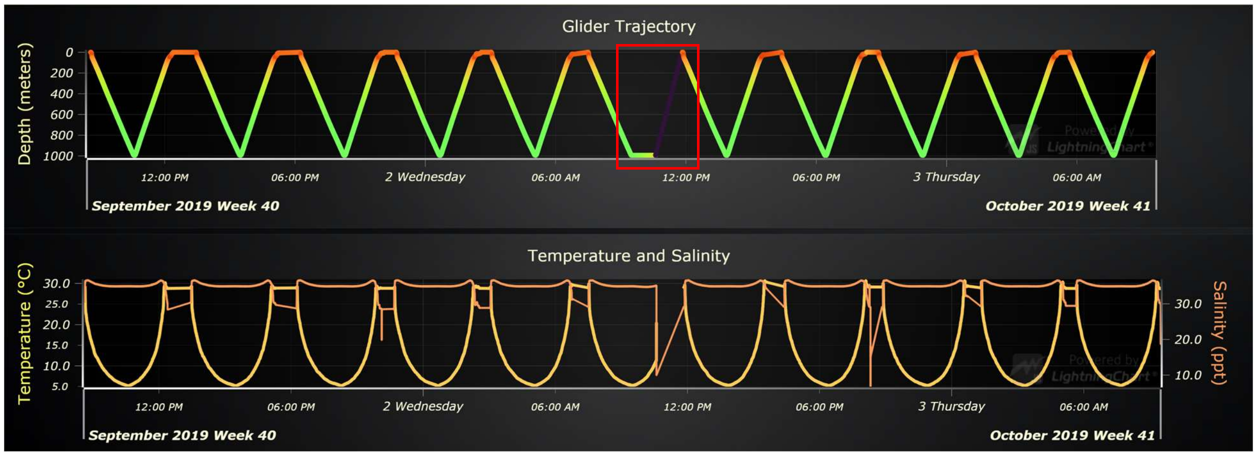


Figure 7: Measurement errors in glider data observed using "ctd_depth" variable for glider depth.

Figure 8 uses the “depth” variable for glider depth. Gaps in the depth data are observed indicating that the depth was either being incorrectly measured or incorrectly timestamped. This trend in depth measurements continues through the rest of analyzed data, from dive 45 to 63. Because of this observation, the “ctd_depth” variable was used in the application.

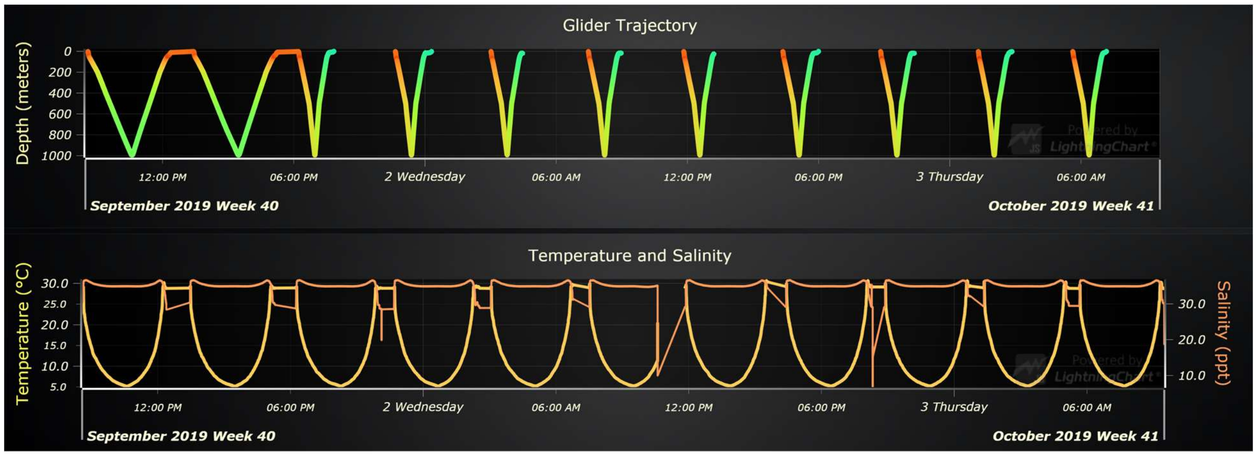


Figure 8: Gaps in depth data on Glider Trajectory plot using "depth" variable for glider depth.

Web Application Development Method

Application Framework

The Glider Data Viewer app utilizes modern web application development technology for easy online deployment. The application framework was set up using Nuxt.js 2, an open source JavaScript framework that streamlines the creation of web applications. Nuxt.js utilizes Vue.js, Node.js, Webpack, and Babel.js. The Vue.js framework provides quick and simple front end web application development that is sleek and performant for the end user. Webpack packages all the development files together in a way that is efficient for loading in the browser. Babel.js transpiles the code into a backwards compatible version for the browser. The creation of the Nuxt.js application requires the use of npm (Node.js package manager), a command-line utility that installs and manages open source software packages from the npm.js online JavaScript repository.

Development Environment Setup

For running the application on a local server, or for future development of the Glider Data Viewer, the development environment can be replicated and the code can be downloaded and installed from the GitHub code repository. An understanding of HTML, JavaScript, CSS, and Vue.js is recommended for development. The application utilizes Vue.js version 2 syntax.

Node.js version 16.17.0 and npm version 8.17.0 was used in the development of this application. Development of the app requires a text editor and terminal. Visual Studio Code is recommended which provides an editor with a built-in terminal. The following VS Code extensions are also recommended: Babel, ESLint, Prettier, Vetur, Vue. For more detailed information about the software dependencies and versions used in the application, see the package.json in the code repository. These packages will be automatically installed by npm when installing the application.

Download VS Code or your preferred text editor and terminal. Install Node.js and npm. The following website guides the user through the process, https://docs.npmjs.com/downloading-and-installing-node-js-and-npm. Clone the repository from GitHub, https://github.com/awalsh64/glider.

```
git clone https://github.com/awalsh64/glider.git
```

Run the following command to install the application.

```
npm install
```

To create a local server to deploy the application for local use or development testing, run the following command and open localhost:3000 in a web browser.

```
npm run dev
```

Other users can configure their GitHub Pages to deploy the app and their future changes using the GitHub Pages Quickstart Guide, https://docs.github.com/en/pages/quickstart. The app has been configured with the push-dir package to deploy to a “gh-pages” branch with the following commands [3].

```
npm run generate
```

```
npm run deploy
```

Once the commands are run, set your GitHub Pages target deployment branch to “gh-pages” in your GitHub repository [3].

Recommended Future Work

A new method for handling the processing of the audio file chunks for the getByteFrequencyData is available, but at the time of development, there was inadequate documentation available for learning how to implement this process. If future development is done, the ScriptProcessorNode is recommended to be converted to an AudioWorkletNode [9].

If future development of the Web Audio API introduces the ability to change the windowing functions, this feature can be added to the FFT Parameters selections. The development of this feature is not yet schedule, but can be tracked on GitHub, https://github.com/WebAudio/web-audio-api/issues/2421.

Displays for additional NetCDF variables can be created if needed. The capability for reading the NetCDF variables and looking at all the attributes is possible with the current netcdfjs package, so a user interface could be added to the app to look at the data [8]. A flexible plot could also be developed that allows the user to select any NetCDF variables to plot against each other.

Overlays for additional environmental variables, such as sea surface temperature or chlorophyll, could be added to the Geo Location plot.

Modern web development technologies are constantly improving. This report explains a proof-of-concept method for using the Web Audio API for acoustic audio processing, but this method can be optimized. The use of different technologies or development methods could be used in future iterations of this application to produce faster or less memory intensive processing capabilities.

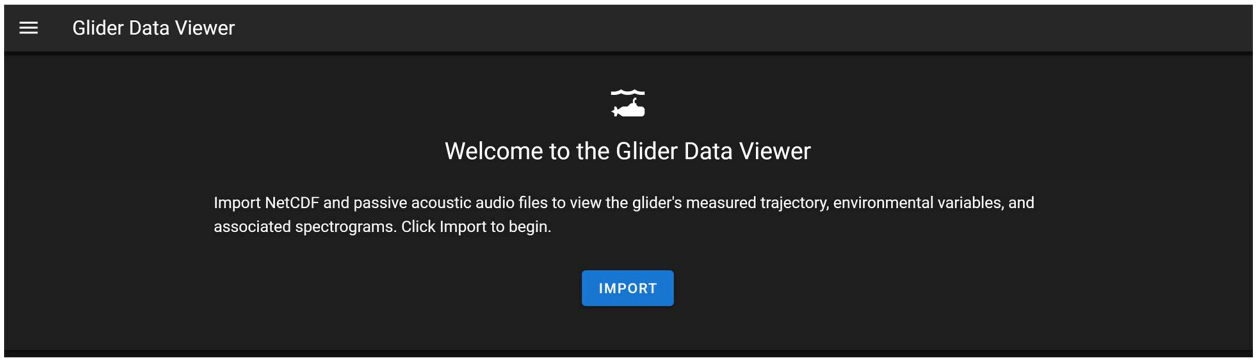
Sources

1. “Arction/Lcjs-Vue-Template: Vue Project Template with Lightningchart JS.” *GitHub*, https://github.com/Arction/lcjs-vue-template.
2. “BaseAudioContext.createScriptProcessor.” *Web APIs*, Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/BaseAudioContext/createScriptProcessor.
3. *Deploy Nuxt JS on Github Pages*, https://elpan.dev/en/deploy-nuxt-js-on-github-pages.
4. “Free JS Charts Library for Non-Commercial Use by LightningChart.” *LightningChart*, 24 Oct. 2022, https://lightningchart.com/js-charts/community-license/.
5. “High-Performance JavaScript Charts - LightningChart® JS.” *LightningChart*, 2 Nov. 2022, https://lightningchart.com/js-charts/.
6. “JavaScript Heatmap Spectrogram Chart.” *LightningChart*, Arction, https://lightningchart.com/lightningchart-js-interactive-examples/examples/lcjs-example-0802-spectrogram.html.
7. K.V. Mackenzie, Nine-term equation for the sound speed in the oceans (1981) J. Acoust. Soc. Am. 70(3), pp 807-812.
8. *Netcdfjs 2.0.2 Documentation*, https://cheminfo.github.io/netcdfjs/.
9. “ScriptProcessorNode.” *Web APIs*, Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/ScriptProcessorNode.
10. *Web Audio API*, W3C, 7 Nov. 2022, https://webaudio.github.io/web-audio-api/#AnalyserNode-methods.
11. “Web Audio API.” *Web APIs*, Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API.

Glider Data Viewer User Guide

Homepage

When first accessing the Glider Data Viewer app, you will be brought to the homepage.

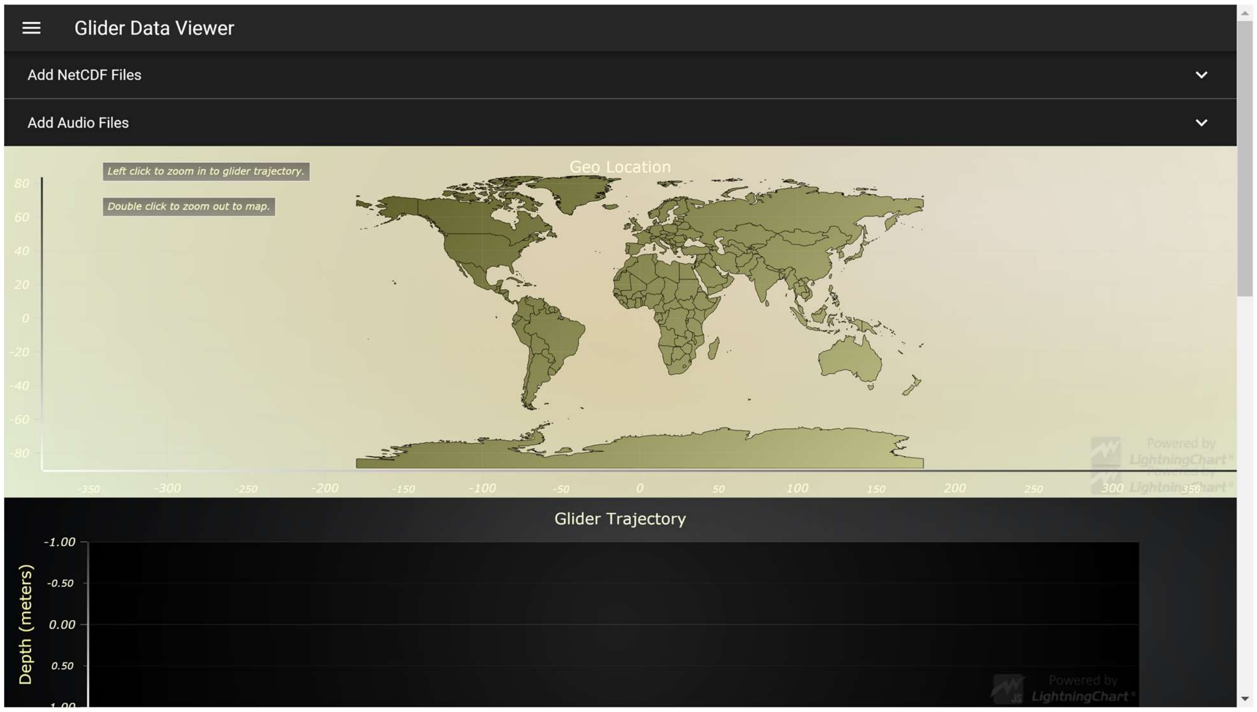


Welcome to the Glider Data Viewer

Import NetCDF and passive acoustic audio files to view the glider's measured trajectory, environmental variables, and associated spectrograms. Click Import to begin.

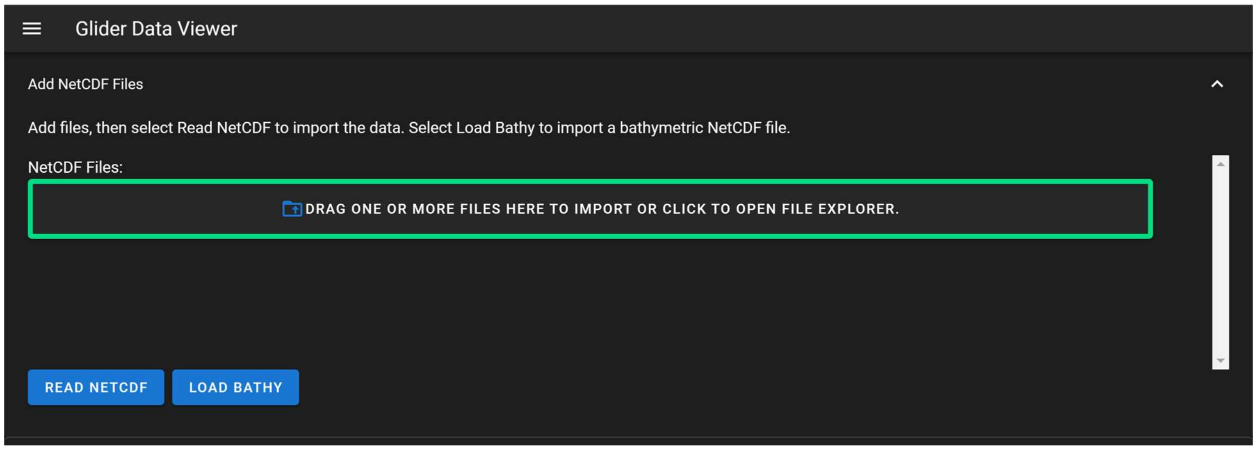
Import Page

Upon clicking Import, you will be brought to the main app screen, the Import page.



Add NetCDF Files

Clicking the first expansion drawer will allow you to load NetCDF files from the glider. The files must include the following variables: “ctd_time”, “ctd_depth”, “temperature”, “salinity_raw”, “latitude”, and “longitude”. Drag the files into the green box, or click the box to open the file explorer and select files. Once selecting the desired files, select the Read NetCDF button to import the data.



Glider Data Viewer

Add NetCDF Files

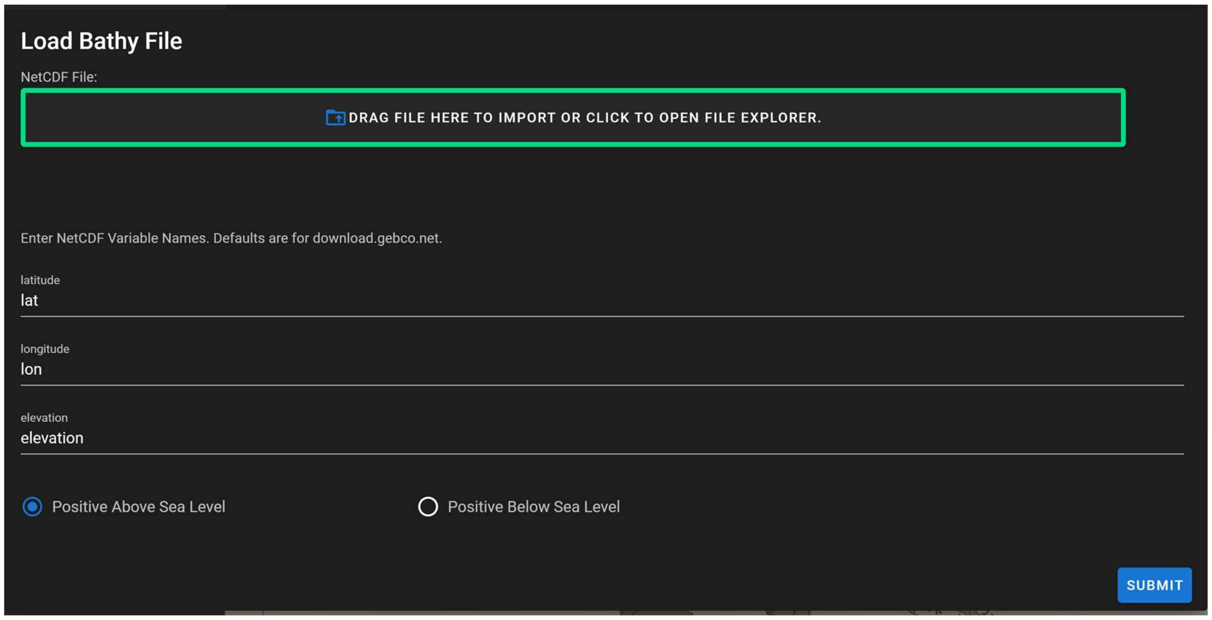
Add files, then select Read NetCDF to import the data. Select Load Bathy to import a bathymetric NetCDF file.

NetCDF Files:

DRAG ONE OR MORE FILES HERE TO IMPORT OR CLICK TO OPEN FILE EXPLORER.
READ NETCDF
LOAD BATHY

Load Bathy

Clicking the Load Bathy button will bring up the Load Bathy File window. Drag a NetCDF file of bathymetric data into the green box, or click the box to open the file explorer. The file must contain variables for latitude, longitude, and elevation or depth at each latitude, longitude point. Enter the names for the variables if they are different than the default. Select Positive Above Sea Level for elevation and Positive Below Sea Level for depth. The defaults are set for the GEBCO Gridded Bathymetry Data Download 2D NetCDF Grid, available at https://download.gebco.net/. Select Submit to load the file.



Load Bathy File

NetCDF File:

Enter NetCDF Variable Names. Defaults are for download.gebco.net.

latitude: lat

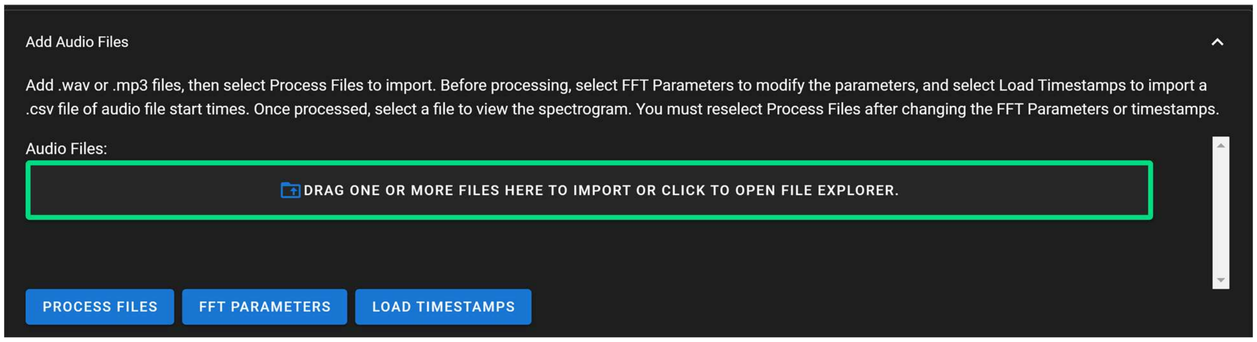
longitude: lon

elevation: elevation

[✓ Positive Above Sea Level] Positive Below Sea Level

Add Audio Files

Clicking the Add Audio Files expansion drawer will allow you to load .wav or .mp3 audio files. The audio files must follow the timestamp naming convention to properly display the timestamps on the plots. The file name must end with "yymmdd_hhmmss.wav" (or .mp3) where the date and time specifies the start time of the audio file and the hours are measured out of 24. The required file name values can be preceded with any number of characters.



Add Audio Files

Add .wav or .mp3 files, then select Process Files to import. Before processing, select FFT Parameters to modify the parameters, and select Load Timestamps to import a .csv file of audio file start times. Once processed, select a file to view the spectrogram. You must reselect Process Files after changing the FFT Parameters or timestamps.

Audio Files:

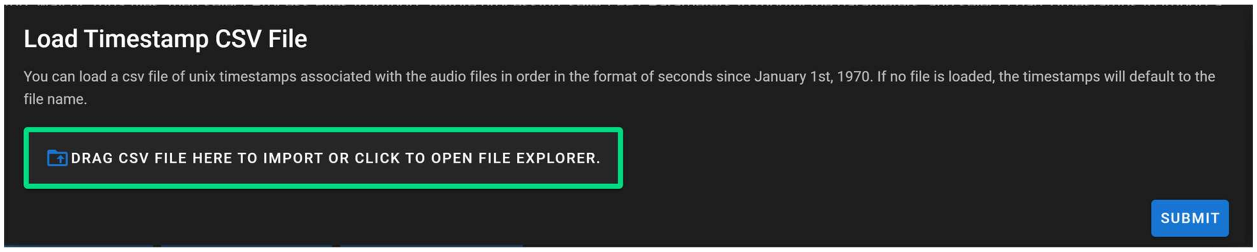
DRAG ONE OR MORE FILES HERE TO IMPORT OR CLICK TO OPEN FILE EXPLORER.
PROCESS FILES
FFT PARAMETERS
LOAD TIMESTAMPS

Process Audio Files

Once the audio files are selected, click the Process Files button to create the spectrograms. Processing and displaying the audio files with high resolution is a memory intensive process that will fail if the computer's available memory is exceeded. The length of audio that can be processed and the resolution to which it can be displayed is dependent on the available memory of the computer used.

Load Timestamps

If the audio file is not named with the correct timestamp naming convention, a .csv (comma separated values) file of the timestamps can be imported from the Load Timestamps button. The .csv file must have a column of timestamps that corresponds to the audio files loaded in order. The timestamps must be in Unix timestamp format of seconds since January 1st, 1970. The timestamp file must be loaded before the audio files are processed, or you must process the files again.



Load Timestamp CSV File

You can load a csv file of unix timestamps associated with the audio files in order in the format of seconds since January 1st, 1970. If no file is loaded, the timestamps will default to the file name.

DRAG CSV FILE HERE TO IMPORT OR CLICK TO OPEN FILE EXPLORER.
SUBMIT

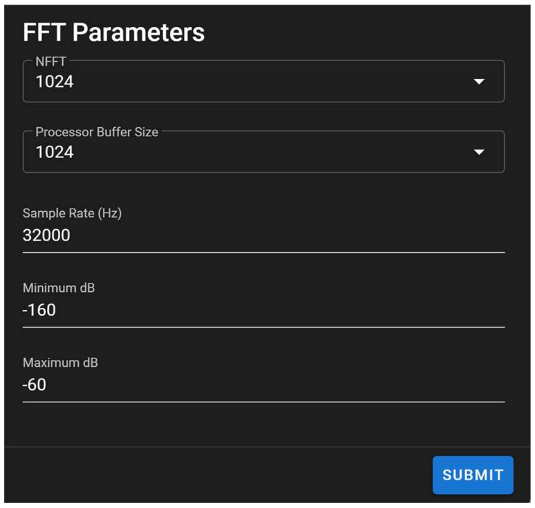
FFT Parameters

Selecting the FFT Parameters button will bring up the selection window where the processing and display parameters can be edited. The user can define the parameters for performing the Fast Fourier Transform (FFT) on the audio files to adjust the frequency, time, and amplitude ranges and resolutions. The FFT parameters must be set before the audio files are processed, or you must process the files again.

The NFFT parameter sets the number of frequency points used to calculate the FFT and controls the resolution in the frequency domain. The Processor Buffer Size parameter specifies the length of data chunks provided to the frequency analyzer and controls the resolution in the time domain. The NFFT and Process Buffer Size values can be adjusted to balance the resolution and the processing time and power required by the computer. They must be a power of two between 256 and 16384.

The Sample Rate parameter is defined in hertz and can be in the range of 3000 to 768000 Hz. It should be set to match the sample rate of the audio file, or down sampled to decrease processing time, but will cause a failure if set greater than the audio file sampling rate. The sample rate controls the maximum frequency of the spectrogram, which is equal to half the sample rate.

The Minimum dB and Maximum dB parameters specify the scaling range for the amplitude of the output data. The Maximum dB value must be less than or equal to 0 and greater than -100dB. The Minimum dB value must be less than the Maximum dB value. The amplitude colormap for the spectrogram will be adjusted to match the decibel range.


FFT Parameters

NFFT
1024

Processor Buffer Size
1024

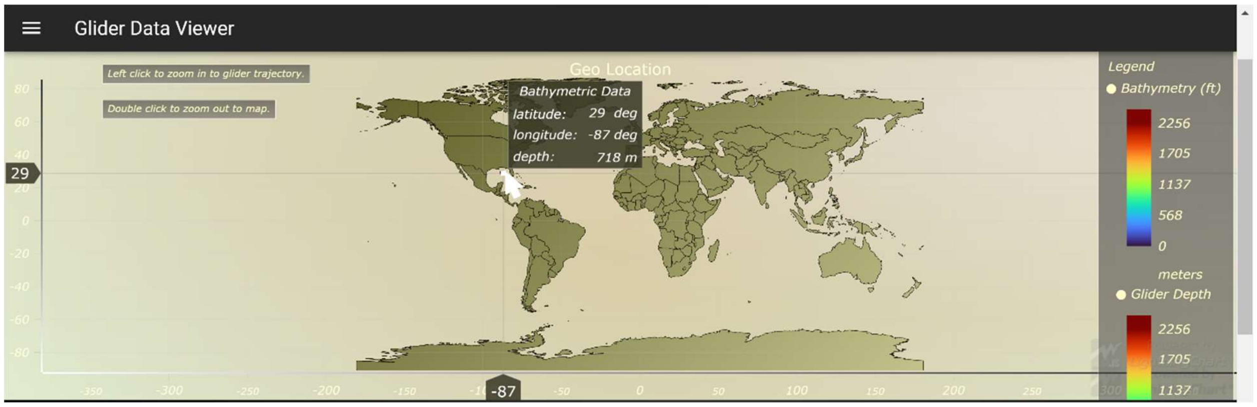
Sample Rate (Hz)
32000

Minimum dB
-160

Maximum dB
-60

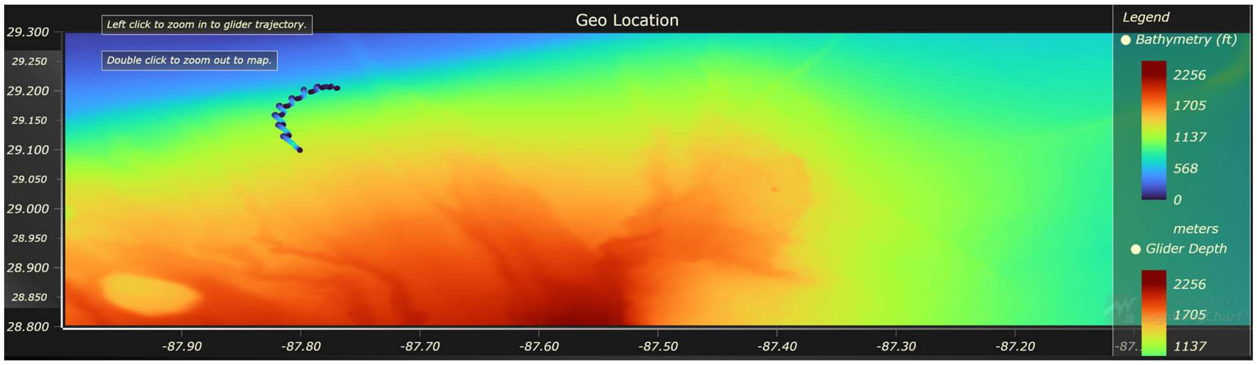
Geo Location Map Plot

Once NetCDF files are loaded, a dot appears on the Geo Location plot in the location of the data. Hovering over the plot brings up a tooltip with the latitude, longitude, and bathymetric depth if hovering over a bathymetric point, or latitude, longitude, glider depth, and glider speed if hovering over a glider point.



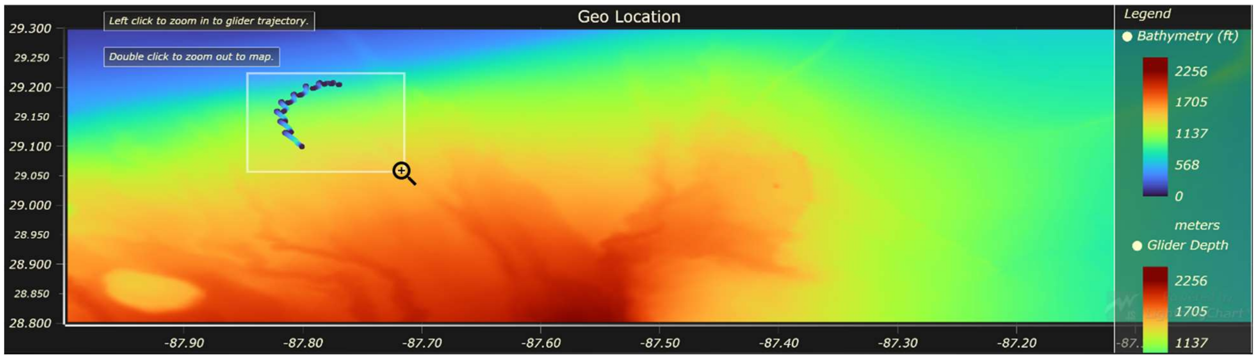
Geo Location Bathymetry and Trajectory Plot

Left clicking once on the map of the Geo Location plot will zoom in to fit the imported data. The glider path is overlayed on the bathymetric colormap. The glider path is colormapped for the glider depth. The legend is the same for both the bathymetric depth and the glider depth. Double click the zoomed in plot to zoom back out to the full world map view.

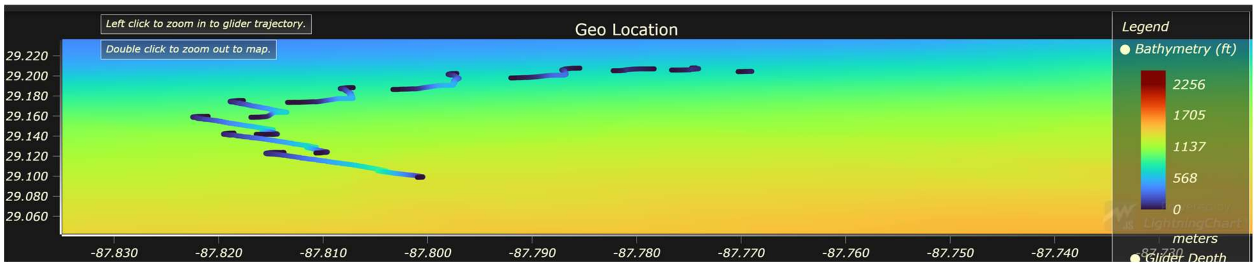


Plot Zoom In Controls

Click and drag from left to right to select a box to zoom in further on the Geo Location plot.

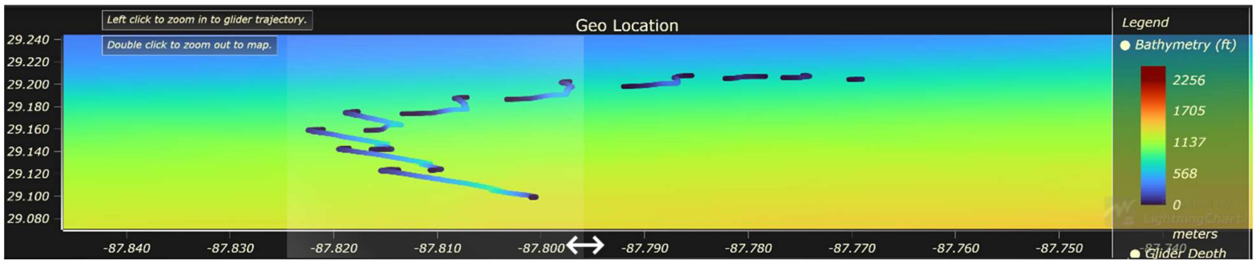

Geo Location
Left click to zoom in to glider trajectory.
Double click to zoom out to map.
Legend
Bathymetry (ft)
2256
1705
1137
568
0
meters
Glider Depth

The plot will zoom in to fit the window to the selected box.

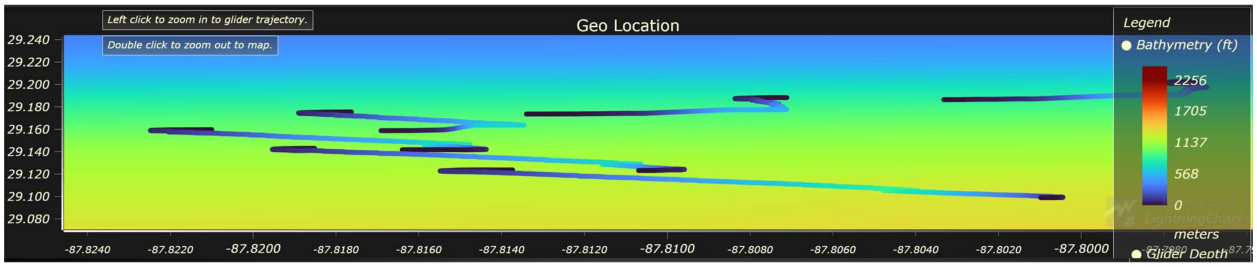


Axes Zoom Controls

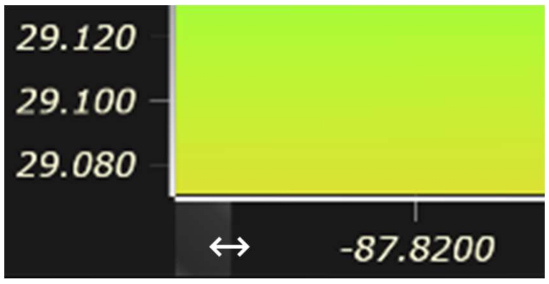
Clicking and dragging on the x or y axis selects the bounds of that axis to zoom in to.



The plot will zoom in on just the selected axis.



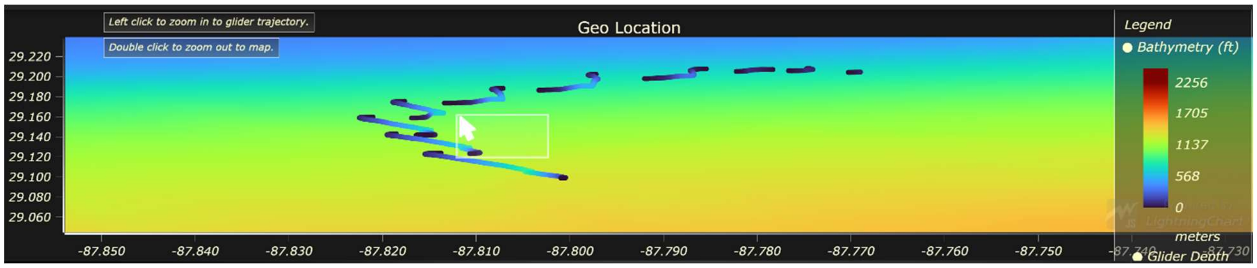
Hovering over either edge of the x or y axis will bring up a transparent box. Clicking here allows you to drag in or out to adjust that axis bound.



Scrolling with the mouse scroll wheel while hovering over an axis will also zoom in or out on that axis.

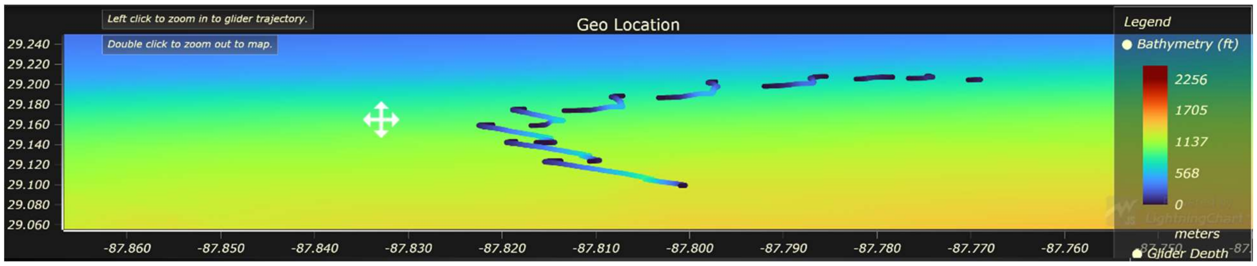
Plot Zoom Out Controls

Clicking and dragging from right to left anywhere on the plot will zoom back out to fit all the data.



Plot Panning Controls

Right clicking on the plot and dragging pans the area of the plot.

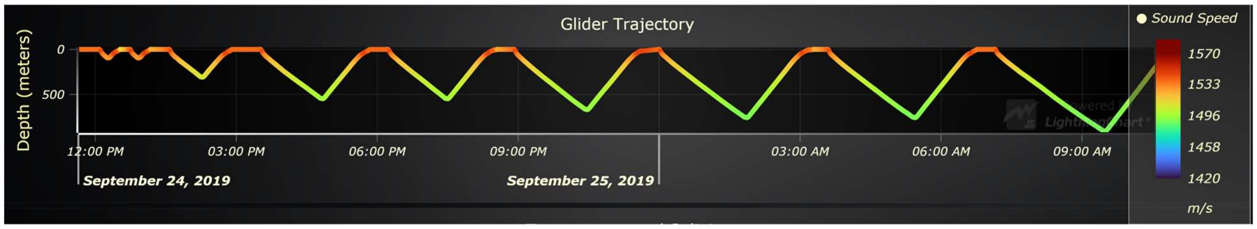


Legend Controls

The Legend can be dragged if it is covering a part of the plot that is being analyzed. Clicking on a legend entry will turn off that data series.

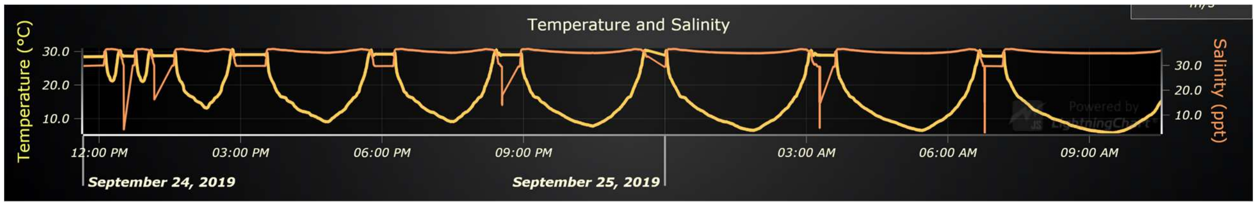
Glider Trajectory Plot

The Glider Trajectory plot displays the depth of the glider over time with a colormap for the sound speed at each location. The zoom features detailed above for the Geo Location plot are also available for the Glider Trajectory Plot.



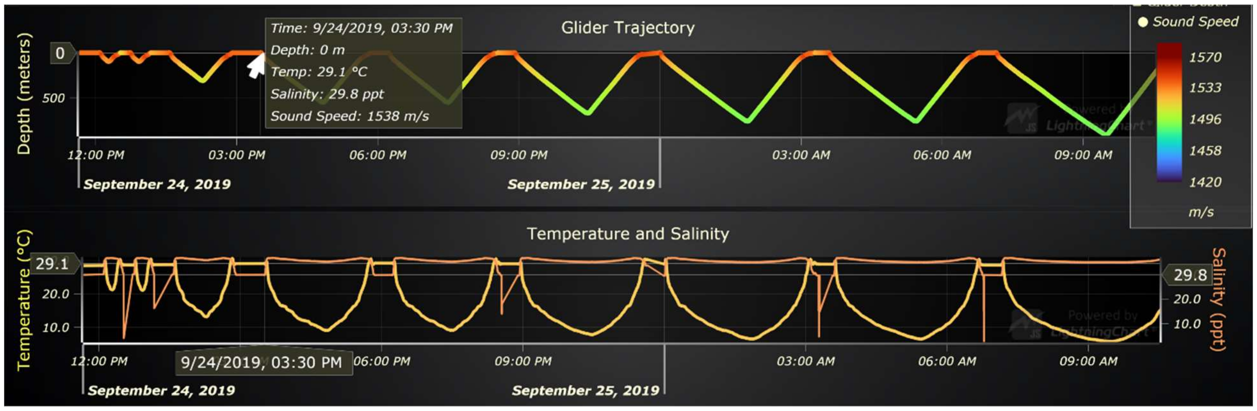
Temperature and Salinity Plot

The Temperature and Salinity plot displays the temperature in yellow on the left y axis and salinity in orange on the right y axis over time. The zoom features detailed above for the Geo Location plot are also available for the Temperature and Salinity Plot.



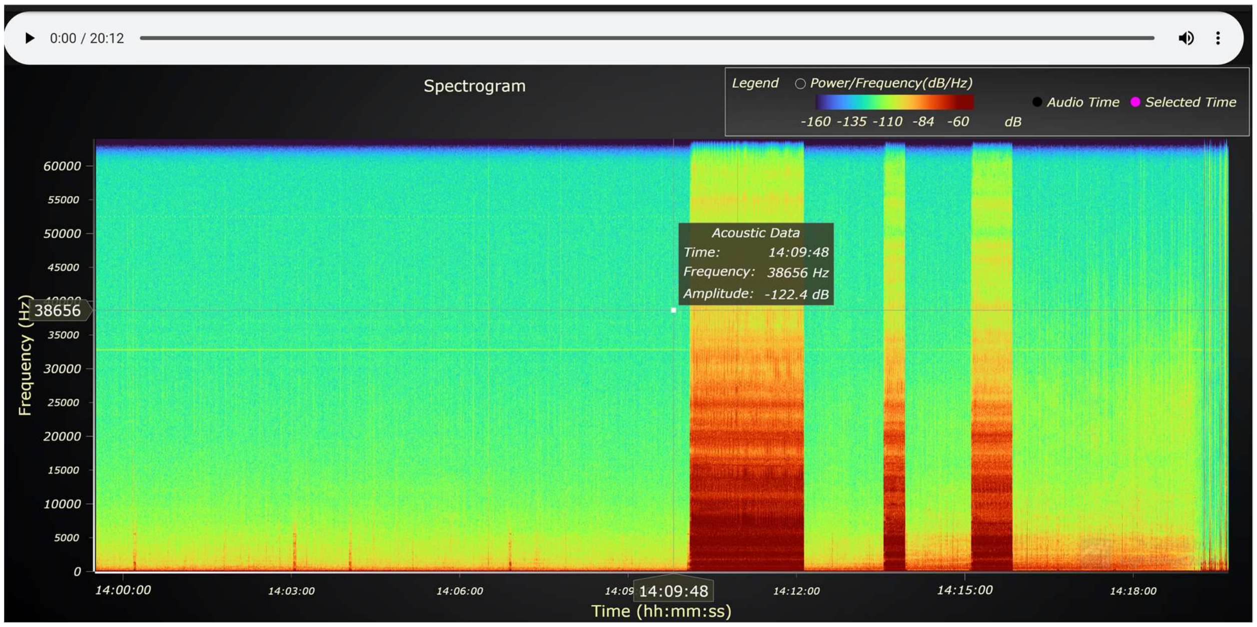
Environmental Tooltips

The Glider Trajectory and Temperature and Salinity plots are linked so that when hovering over one will show the tooltips and axis markers for all axes of both plots. The tooltips provide the timestamp, glider depth, temperature, salinity, and sound speed.



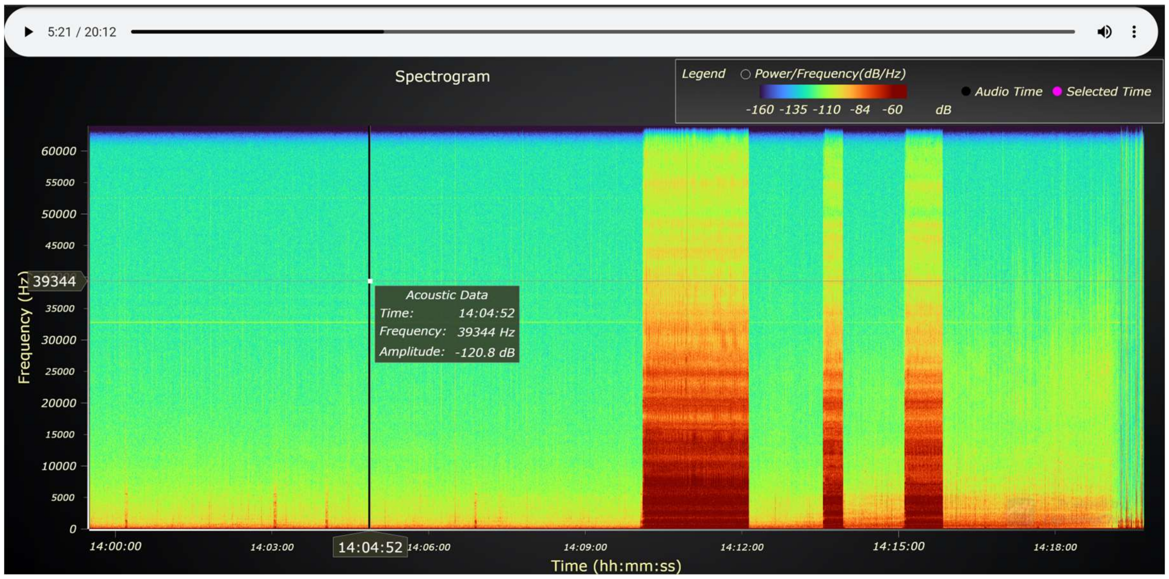
Spectrogram Plot

The Spectrogram Plot displays the amplitude of the frequencies over time. The plot will display the spectrogram selected by clicking the glider plots, or by selecting the file in the Audio Files list. The zoom features detailed above for the Geo Location plot are also available for the Spectrogram Plot.



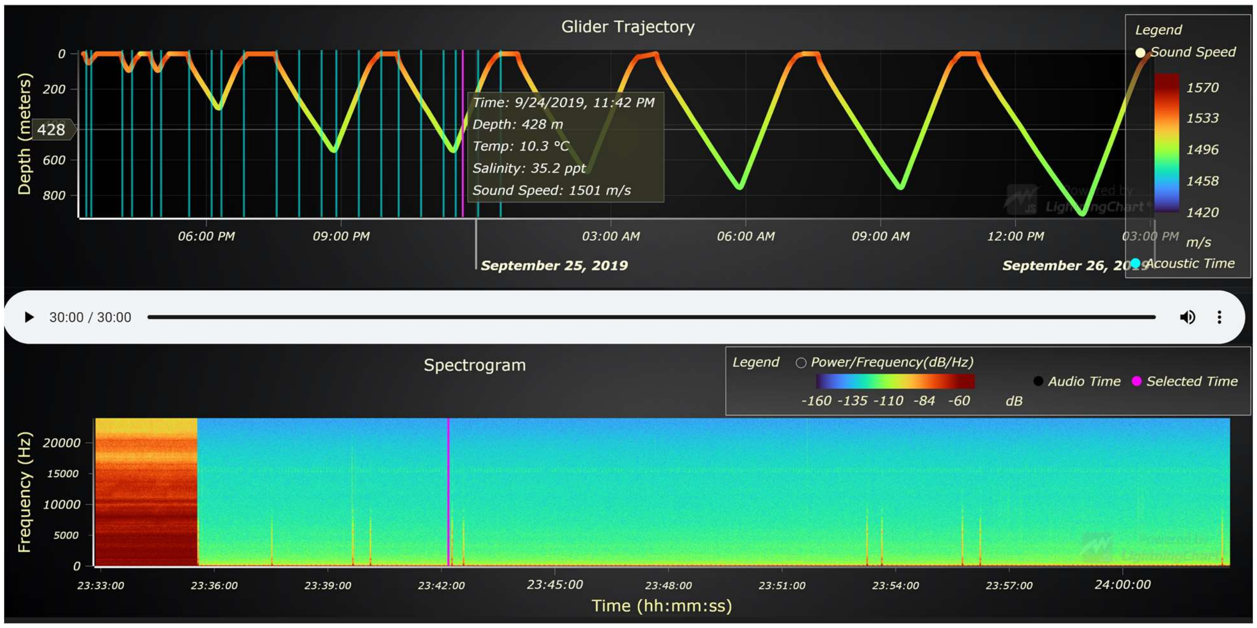
Audio Playback

When playing the audio, a black line on the Spectrogram plot indicates the current playback time of the audio. Clicking the spectrogram plot will move the audio playback time to the selected time. Depending on the resolution of the spectrogram, the audio playback time marker may only be accurate within a few hundred milliseconds due to the resolution of the HTML audio player.



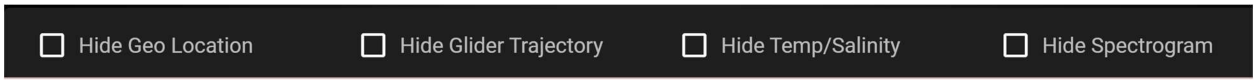
Audio File Times

When audio files are loaded, cyan lines will display on the Glider Trajectory and Temperature and Salinity plots indicating the start times of the audio files. Clicking on the plots will load the spectrogram that starts before the selected time. The selected time will be indicated with a magenta line.



Plot Layouts

At the bottom of the page, there are toggles for each of the plots that allow the user to manipulate the layout of the page by hiding certain plots. If both the Glider Trajectory and the Temperature and Salinity plots are hidden, the Spectrogram display will be full screen. If either of the glider plots are shown, the Spectrogram display will be half screen to allow viewing of two plots at once. Pressing the F11 key will put the application into full screen, allowing more visual space for the displays.


Hide Geo Location
Hide Glider Trajectory
Hide Temp/Salinity
Hide Spectrogram