

CASTool User Guide – outline

1. Background
 - a. Why developed
 - i. CWA requirements
 - ii. Stressor ID process/weight-of-evidence/CADDIS
 - iii. Need for more rapid methods/tools
 - b. 1st CASTool iterations – AZ and SD
 - c. Goal of current effort is more broadly applicable tool that all states can use
2. CASTool overview
 - a. Overview of how it works
 - i. Corresponds to Stressor ID steps
 - ii. Compares test site values to comparator sites, for any biological community
 - iii. Primary index, other metrics
 - iv. Input data
 - v. Clustering
 - vi. Report generated
 - b. Inputs/outputs figure/table
 - c. Lines of evidence table (without scoring details here?)
 - d. Skeleton code and Shiny app
 - i. What developed in
 - ii. Where available
 - e. Caveats/limitations
 - i. Screening level tool
 - ii. Not meant to spit out answer that you accept without looking at what into answer
 - iii. Garbage in, garbage out – depends on quality of data put into tool
3. Deep dive into how CASTool code/Shiny app works
 - a. Key definitions – test site, comparator vs. non-comparator sites, index vs. metrics
 - b. Standard figure features – scoring at top, how test site values are delineated
 - c. Lines of evidence and scoring
 - d. Table summarizing default values that you can change
 - e. Walk through app with screenshots
4. Input data preparation
5. Contact us with questions/feedback

List of input files

List of fields/descriptions in each file

Lines of evidence crosswalk with CADDIS?

Example – data files, workflow, report, interpretation

List of clustering variables

Commented [KS1]: Do we need to do both code and app here? Or maybe handle code in appendix?

Call-out boxes for key reminders:



Summary boxes for start of each section:

On this page, you can:

- TEXT
- TEXT

1. Background

2. CASTool overview

2.1 In brief

a. Overview of how it works

- i. Corresponds to Stressor ID steps
 - ii. Compares test site values to comparator sites, for any biological community
 - iii. Primary index, other metrics
 - iv. Input data
 - v. Clustering
 - vi. Report generated
- b. Inputs/outputs figure/table
 - c. Lines of evidence table (without scoring details here?)
 - d. Skeleton code and Shiny app
 - i. What developed in
 - ii. Where available
 - e. Caveats/limitations
 - i. Screening level tool
 - ii. Not meant to spit out answer that you accept without looking at what into answer
 - iii. Garbage in, garbage out – depends on quality of data put into tool

2.# Definitions

region: the geographic area (e.g., a state) that encompasses the target and comparator sites to be analyzed

metric: biological community response variables

index: the biological community response metric used to determine whether a sample is impaired

3. Shiny application

4. Input data preparation

Templates for CASTool input files are available for download from the ### tab of the Shiny application. Input files are provided to the CASTool in a zipped folder.

A key required input file is _CASTool_Metadata.xlsx, which defines many of the parameters used to set up a CASTool run, including the names of other input files. The metadata file template describes

the function of each metadata variable along with its data type, acceptable values (“Domain”), whether it is required, and default value.

For some regions, some of the required geospatial and cluster input files can be loaded automatically to CASTool from helper packages (see [list of available regions](#)). If your region is not available in the helper package, you can generate these files using the CustomBoundary package in R.

The CASTool currently supports analysis of three biological communities: benthic macroinvertebrates (bmi), algae (alg), and fish. Data for at least one biological community must be provided to the CASTool.

4.# Geospatial and cluster file generation with a custom boundary

To generate the required geospatial and cluster files using a custom boundary, you will need a geospatial file containing the region boundary, a folder to save outputs, and a region name. The CustomBoundary package uses `st_read()` to read in the geospatial file, which can accept a variety of common geospatial file types like shapefile and GeoJSON.

First install the CustomBoundary package using the code below. Uncomment the first line if you do not have the “pak” package installed.

```
# install.packages("pak")
pak:::pak("laura-naslund/CustomBoundary")
```

Copy the code below and replace the example inputs with 1) the path to your boundary file, 2) the path to your desired output folder, and 3) the name of your region. You can copy file paths to your clipboard by right clicking on the file name in your file explorer and selecting: Copy as path (Figure #). If you are using a Windows machine, you will need to replace the backslashes (\) in the pasted file path with forward slashes (/). Ensure file paths and the region name are wrapped in straight quotes ("").

```
CustomBoundary:::generateFiles(inputFilePath =
"C:/Users/UserName/Profile/Desktop/example_region.shp", outputFolder =
"C:/Users/UserName/Profile/Desktop/Output", region = "exampleRegion")
```

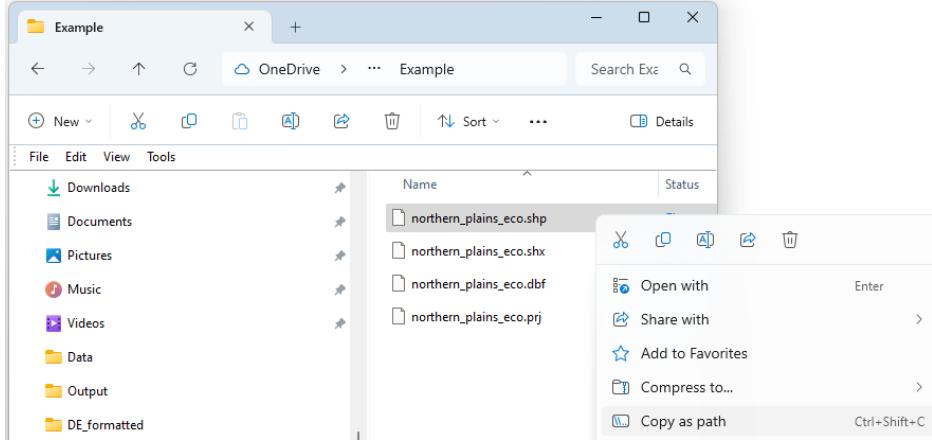


Figure 1: To copy a file path to your clipboard, right click the file name in your file explorer and select: Copy as path.

When generateFiles() has finished running, you will find the required input files in the specified output folder. Copy these files into the input data folder you intend to zip. Add the names of the resulting files to the corresponding parameters in the _CASTool_Metadata.xlsx (Figure #). Note that the ClusterOutput folder contains additional files generated by the clustering algorithm that are not required as CASTool inputs. [Maybe additional elaboration that if a user wants something other than the default number of clusters, those outputs can be found here].

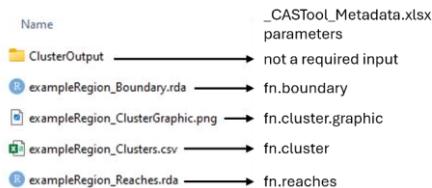


Figure 2: Output files generated by the CustomBoundary package and their corresponding metadata parameter names.