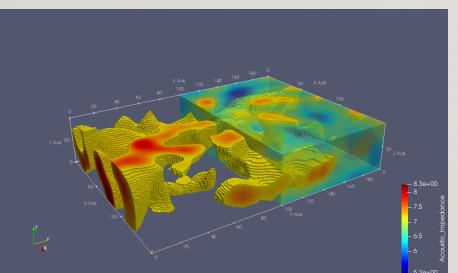
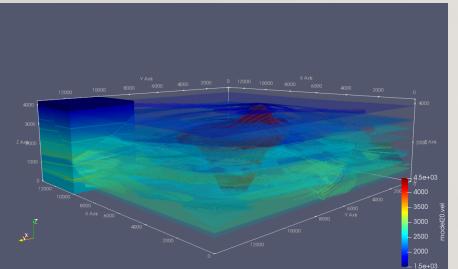




# Illuminating the Value of Geophysical Imaging Through Visualization and Virtual Reality

---



BANE SULLIVAN

# INTRODUCTION

---

- The goal of this project is to develop a framework to funnel geophysical data/models into virtual reality
  - Extract Value of Information (VOI)
  - User/Stakeholder engagement with geophysical findings
- My specific goal is to develop a heavily documented library of plugins, macros, and examples of how to view standard formats of geoscientific and geophysical data on the ParaView software platform.
  - ParaView: open-source, multi-platform, data analysis, and visualization application by Kitware (<https://www.paraview.org>)
- These plugins will provide tools to perform post processing visual analysis and interpretation of geoscientific data and models
- All plugins wiki documentation, and tutorials are published on GitHub:  
(<https://github.com/banesullivan/ParaViewGeophysics>)
- Accepting end-user feature requests on the repo's Issues page

# WHY ParaView



- 
- Free!
  - Open-source and multi-platform
  - Handles large data sets with ease and fluidity
  - Processing pipeline – reproducible and easy to alter visualizations of complex scenes
  - 3<sup>rd</sup> Party Plugins
  - **Python** VTK wrapping, scripting, and programmable plugins
    - Easy to make your own plugins and data format readers
    - Macros for reproducible visualizations of complex scenes
  - Extends into Virtual Reality

# PROJECT DELIVERABLES

---

- Develop and **document** geoscientific plugins for ParaView
- **Document** how to use ParaView's native features to complete common tasks in the visualization of geoscientific data
- Gain an ability to represent 3D spatially referenced data intuitively to interested parties and stakeholders
- Yield value of information in geophysical applications: de-risk decision making
- Develop a framework to build visualizations for the purpose of stakeholder engagement

# APPLICATIONS

---

- Visual data integration
- Informed decision making and model quality control
- Represent 3D spatially referenced data intuitively to interested parties and stakeholders
  - Stakeholder engagement with geophysical findings
- Reference data in relation to intuitive features like topography, well locations, survey points, or other known features
- Transfer visualizations into Virtual Reality for real scale interaction

# VALUE OF INFORMATION

---

- 3D & VR visualizations contribute to minimizing conceptual uncertainty
- 3D spatial visualizations:
  - Help relate features in a way that our brains better understand
  - Convey meaning to non-technical stakeholders
- VR spatial visualizations:
  - Add scale to data that are brains can more easily understand
  - Communicate uncertainty in an useable way
  - Allows us to traverse the 3D scenes in way we cannot on a flat screen
  - Spatial data becomes more real and more actionable

# VOI METRICS THROUGH “GAMES”

---

- We plan to develop metrics for data/model interaction in VR
- VOI Games:
  - Complex scenes for geophysical data/model interaction
  - Players will have to find features of interest in VR vs. on a monitor
    - E.g. How many planned drills miss their target?
  - We will evaluate different users' abilities to make spatial decisions
    - E.g. Should tunneling continue in location given a spatial stress model?

 [banesullivan / ParaViewGeophysics](#) Unwatch  1 Star  0 Fork  0

Code

Issues 0

Pull requests 0

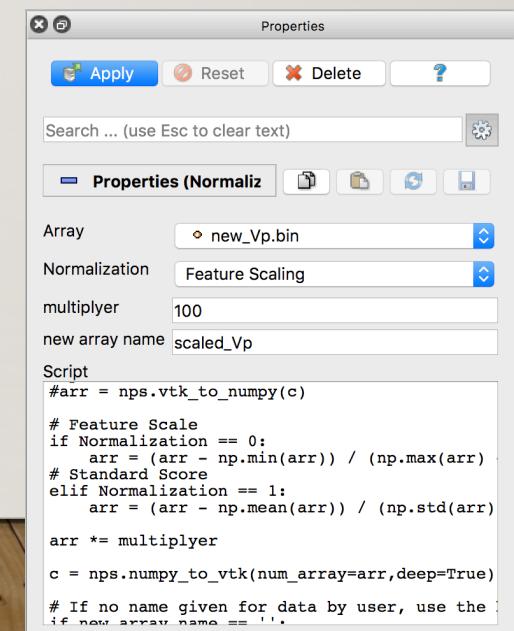
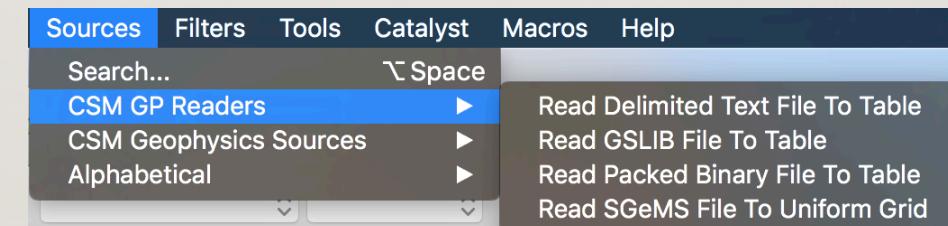
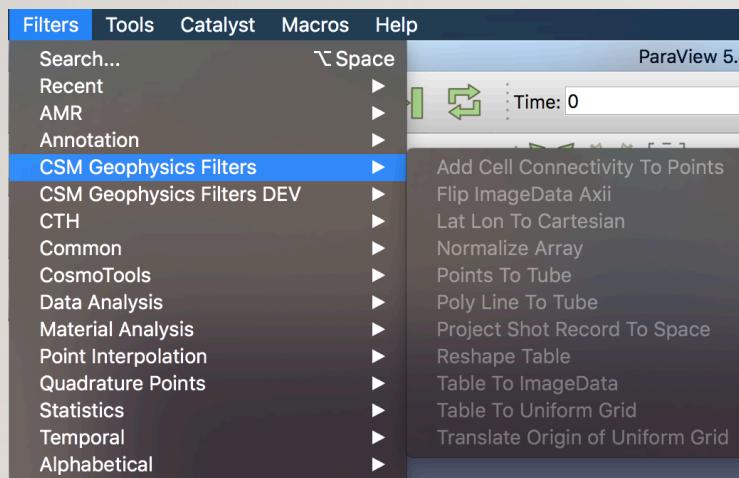
Projects 0

Wiki

Settings

Insights ▾

# Back to Deliverables



# FOCUS ON ParaView PLUGINS

---

- Before we get to VR, we must create the complex 3D scenes we want to see in VR
- ParaView can handle the creation of scenes from large datasets with ease
- ParaView has VR built into one of its releases
- Once we have an ability to get all our different datasets into ParaView, we can immediately send them over to VR
- What good is VR if we can only view a few data types or formats?
- Not everyone has VR gear and we need feedback from users

# PLUGINS: READERS & FILTERS

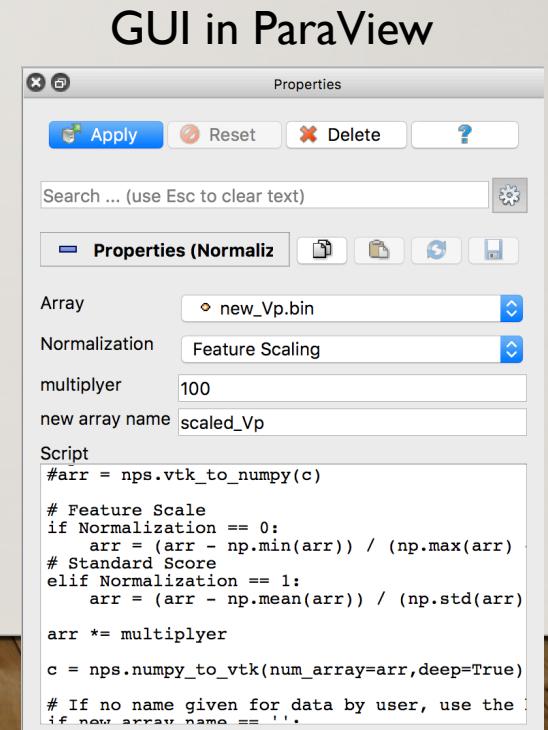
---

- A **reader** puts data from files into proper VTK data structures on the ParaView pipeline
- A **filter** modifies, transforms, combines, analyses, etc. data on the pipeline
- Python Programmable Filters/Readers
  - Easily modified by the end user
  - Able to be wrapped in XML to create a GUI for its use in ParaView
  - Directly edit the source script live in ParaView

Live View of Data

Showing NormalizeArray1 At

	Row ID	new_Vp.bin	scaled_Vp
0	0	1530	0.0925594
1	1	1530	0.0925594
2	2	1530	0.0925594



# EXAMPLE OF SIMPLE READER

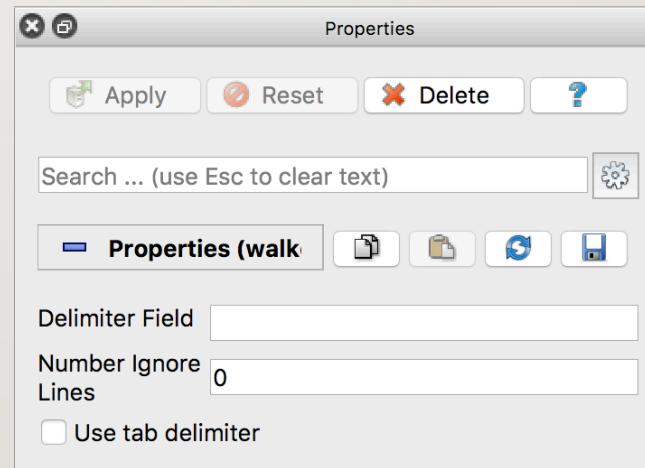
## Customizable Code

```
Name = 'ReadGeoEASFileToTable'  
Label = 'Read GeoEAS File To Table'  
FilterCategory = 'CSM GP Readers'  
Help = 'The GeoEAS file format has headers lines followed by the data as a space  
delimited ASCII file (this filter is set up to allow you to choose any single  
character delimiter). The first header line is the title and will be printed to  
the console. This line may have the dimensions for a grid to be made of the  
data. The second line is the number (n) of columns of data. The next n lines are  
the variable names for the data in each column. You are allowed up to ten  
characters for the variable name. The data follow with a space between each  
field (column).'  
  
NumberOfInputs = 0  
OutputDataType = 'vtkTable'  
ExtraXml = ''  
<Hints>  
...<ReaderFactory extensions="dat geoeas txt"  
...><file_description="GeoEAS File Format" />  
</Hints>''  
  
Properties = dict(  
    FileName='absolute path',  
    Number_Ignore_Lines=0,  
    Delimiter_Field=' ',  
    Use_tab_delimiter=False  
)  
  
def RequestData():  
    import numpy as np  
    import csv  
    from vtk.util import numpy_support  
  
    pdo = self.GetOutput() # vtkTable  
  
    if (Use_tab_delimiter):  
        Delimiter_Field = '\t'  
  
    titles = []  
    data = []  
    with open(FileName) as f:  
        reader = csv.reader(f, delimiter=Delimiter_Field)  
  
        1 Walker data (260x300x1)  
        2  
        3 U  
        4 V  
        5 0 0  
        6 0 0  
        7 0 0  
        8 0 0  
        9 0 0  
        10 0 0  
        11 0 0  
        12 1.686 85.4  
        13 1.771 87.73  
        14 0.023 0  
        15 0.022 0  
        16 0.022 0  
        17 0.017 0  
        18 0 0  
        19 0 0  
        20 0 0  
        21 0 0  
        22 0.019 33.26  
        23 0 0  
        24 0 0  
        25 0 0  
        26 0 0  
        27 0 0  
        28 0.004 0  
        29 0.037 22.94  
        30 0.358 24.57  
        31 18.302 221.5  
        32 16.281 218.67  
        33 32.808 273.49  
        34 17.856 236.35
```

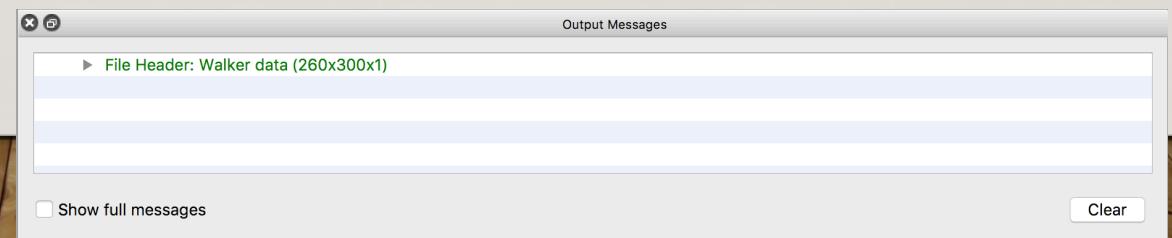
## Input File

```
1 Walker data (260x300x1)  
2  
3 U  
4 V  
5 0 0  
6 0 0  
7 0 0  
8 0 0  
9 0 0  
10 0 0  
11 0 0  
12 1.686 85.4  
13 1.771 87.73  
14 0.023 0  
15 0.022 0  
16 0.022 0  
17 0.017 0  
18 0 0  
19 0 0  
20 0 0  
21 0 0  
22 0.019 33.26  
23 0 0  
24 0 0  
25 0 0  
26 0 0  
27 0 0  
28 0.004 0  
29 0.037 22.94  
30 0.358 24.57  
31 18.302 221.5  
32 16.281 218.67  
33 32.808 273.49  
34 17.856 236.35
```

## GUI in ParaView



## Output to User



## VTK Data

Showing walkerLake.dat			
	Row ID	U	V
0	0	0	0
1	1	0	0
2	2	0	0
3	3	0	0
4	4	0	0
5	5	0	0
6	6	0	0
7	7	1.686	85.4
8	8	1.771	87.73
9	9	0.023	0
10	10	0.022	0
11	11	0.022	0
12	12	0.017	0
13	13	0	0
14	14	0	0
15	15	0	0

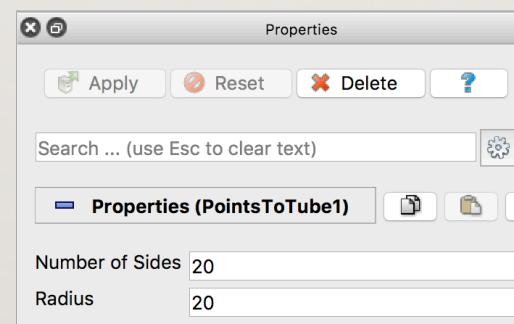
Clear

# EXAMPLE OF SIMPLE FILTER

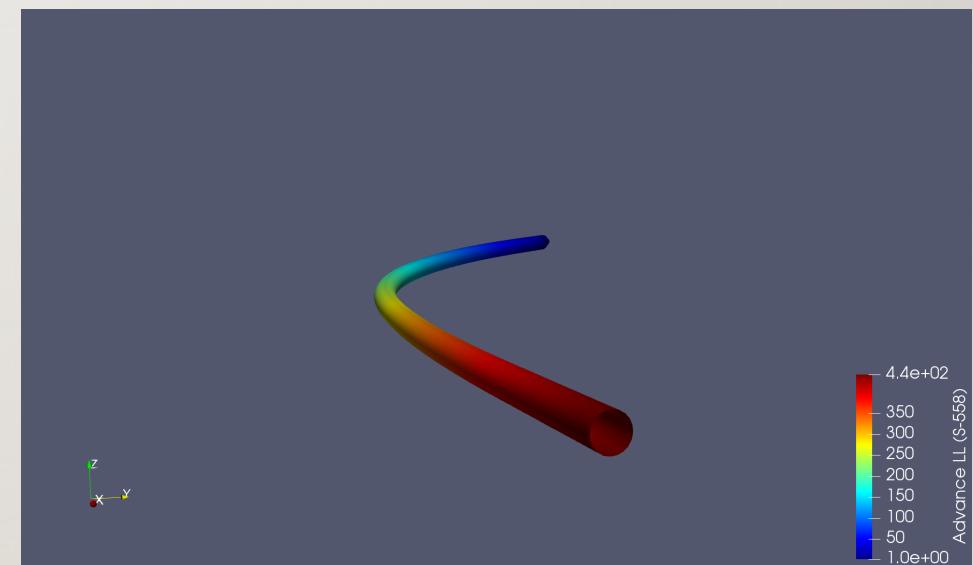
## Customizable Code

```
Name = 'PointsToTube'~  
Label = 'Points To Tube'~  
FilterCategory = 'CSM Geophysics Filters'~  
Help = 'Takes points from a vtkPolyData object and constructs a line of those  
points then builds a polygonal tube around that line with some specified radius  
and number of sides.'~  
  
NumberOfInputs = 1~  
InputDataType = 'vtkPolyData'~  
OutputDataType = 'vtkPolyData'~  
ExtraXml = ''~  
  
Properties = dict(~  
... Number_of_Sides=20,~  
... Radius=10.0,~  
)~  
  
def RequestData():~  
... pdi = self.GetInput() # VTK PolyData Type~  
... pdo = self.GetOutput() # VTK PolyData Type~  
  
... pdo.DeepCopy(pdi)~  
... numPoints = pdi.GetNumberOfPoints()~  
  
... # VTK_POLY_LINE is 4~  
... # Type map is specified in vtkCellType.h~  
... Cell_Type = 4~  
... ptsi = [i for i in range(numPoints)]~  
... pdo.InsertNextCell(Cell_Type, numPoints, ptsi)~  
  
... # Make a tube from the PolyData line:~  
... tube = vtk.vtkTubeFilter()~  
... tube.SetInputData(pdo)~  
... tube.SetRadius(Radius)~  
... tube.SetNumberOfSides(Number_of_Sides)~  
... tube.Update()~  
... pdo.ShallowCopy(tube.GetOutput())~
```

GUI in ParaView



Resulting Spatial Model



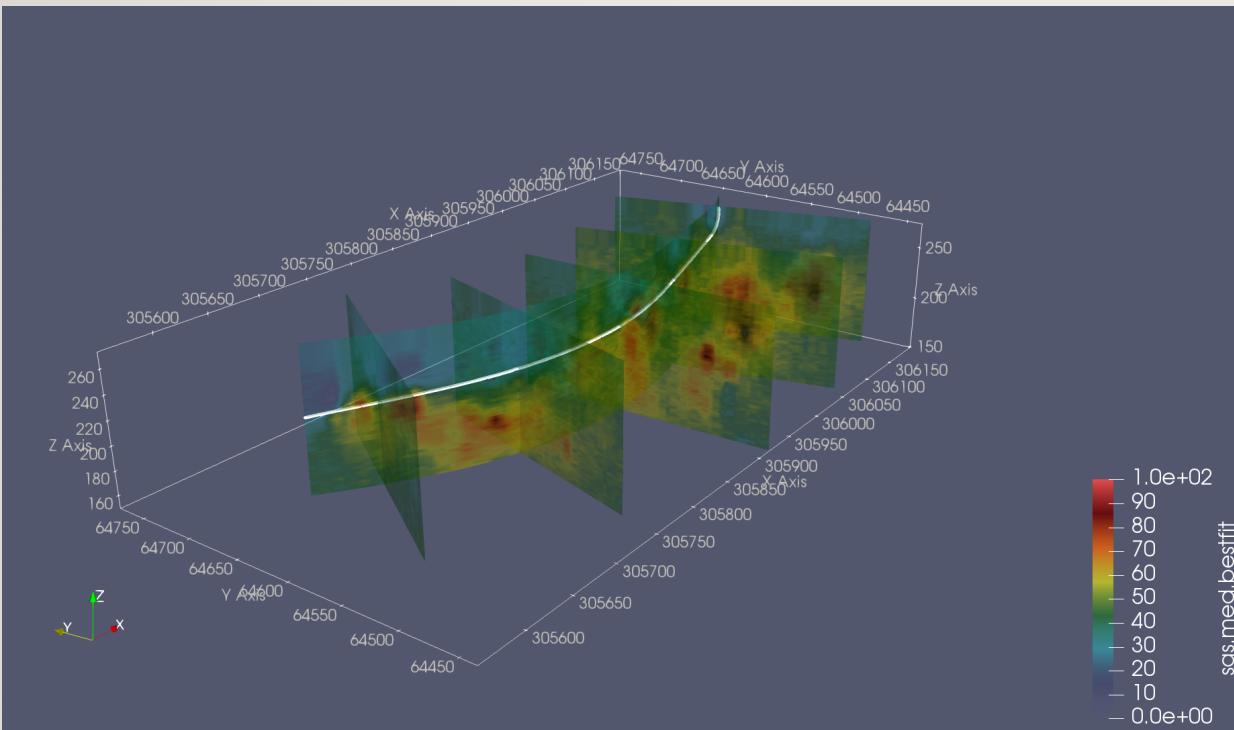
# WORKFLOW MACROS

---

- Macros are Python scripts that complete common or extraneous tasks
- Scripting allows you to create a workflow for a project and run the same complex scene visualizations for various data sets
- Examples:
  - Batch loading of data into ParaView/VTK for visualization
  - Scene navigation: generation of various screenshots
  - Data Slicing: specify a data set and parameters for which you might want to slice it
    - e.g. along some travel path

# CREATING SLICES OF DATA ALONG SPECIFIED ROUTE

---



- We use scripting to do batch processes like generate and save out numerous slices of a model along a specified route
- Or to create a macro to load a complex scene and save animations for various versions of the data

# HOW TO ACCESS THE PROJECT

<https://github.com/banesullivan/ParaViewGeophysics>

The screenshot shows the GitHub repository page for 'banesullivan / ParaViewGeophysics'. The page includes a navigation bar with links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the repository name 'banesullivan / ParaViewGeophysics' is displayed, along with 'Unwatch 1', 'Star 0', and 'Fork 0' buttons. A menu bar below the repository name offers options like 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. The main content area describes the repository as containing plugins for ParaView, tailored for geosciences. It shows statistics: 154 commits, 3 branches, 0 releases, 1 contributor, and BSD-3-Clause license. A commit list is provided, showing recent changes such as renaming files and updating filters.

This repository contains plugins for the open-source, multi-platform data analysis, and visualization application ParaView by Kitware. These plugins are tailored to data visualization in the geosciences with a heavy focus on structured data sets like 2D or 3D mesh grids.

154 commits 3 branches 0 releases 1 contributor BSD-3-Clause

Branch: master New pull request Create new file Upload files Find file Clone or download

Bane Sullivan Renamed file Latest commit b2e03e8 11 days ago

- build Removed in progress filters from master branch. 11 days ago
- colormaps Added more colormaps to use in ParaView 24 days ago
- macros Removed individualistic macros 20 days ago
- src Renamed file 11 days ago
- .gitignore Update so that filters in early stages of development do not get tracked 22 days ago
- LICENSE.md Renamed with .md extension. 2 months ago
- README.md Update syntax for code snippet. 11 days ago
- \_config.yml Set theme jekyll-theme-cayman 2 months ago
- updatePVG.sh Updated how the scripts handle the PVPLUGIN variable and that they pu... 12 days ago

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the right side of the top bar are icons for notifications, a plus sign, and a square with a checkmark. Below the top bar, the repository name 'banesullivan / ParaViewGeophysics' is displayed, along with buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). A horizontal menu bar below the repository name includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki' (which is highlighted with an orange border), 'Settings', and 'Insights'. The main content area features a large heading 'EXPLORE WHAT'S AVAILABLE'.

- Check out the sidebar on the Wiki to see all the currently available features
- All of the readers, filters, and macros developed have a Wiki page explaining:
  - Purpose and what we can apply it to
  - Format specifics, implementation details, parameters, features, etc.
  - Overview of how to use

The sidebar on the right contains the following navigation links:

- Wiki Home
- What's Next
- Wiki Contents
- Getting Started
  - How to Use This Wiki
  - ParaView and VTK
  - Install These Plugins
  - Using 3rd Party Py Libraries
- Plugins
  - About ParaView Plugins
  - Our Readers
  - Our Filters
  - Build Your Own Plugins
- Macros
  - Using ParaView Macros
  - Advice For Making Macros
  - Scene Export Macros
  - Our Macros
- Examples
- Virtual Reality

# STAY UP TO DATE ON WHAT'S NEXT

---

- Check out the “What’s Next” Wiki page for details on upcoming plugins, macros, tutorials, and other information.
- This repo is more than just plugins...
  - Documentation of how to complete common tasks in geoscientific visualization
  - Demos on how to convert specific file formats
  - Demos on how to use macros to make images of complex scenes

The screenshot shows the GitHub interface for the repository 'banesullivan / ParaViewGeophysics'. At the top, there's a navigation bar with links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the right side of the header are icons for notifications, a plus sign, and a user profile. Below the header, the repository name 'banesullivan / ParaViewGeophysics' is displayed, along with statistics: 1 unwatched issue, 0 stars, and 0 forks. A navigation bar below the repository name includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. Further down are buttons for 'Filters', a search bar containing 'is:issue is:open', 'Labels', 'Milestones', and a green 'New issue' button.

# FEATURE REQUESTS

- Anyone can post a feature request on the Issues page of the repo
  - Give your ideas for a filter
  - Request a reader for a specific format
  - Bring to light bugs, errors, and general problems
  - Ask for a tutorial to complete a specific task in data visualization
  - Request macros for different types of scene exports or visualization tasks

# THANK YOU!

---

- Keep an eye out for added code and documentation
- Feedback is essential for this project to be successful (please post on the issues page)
- Request new features!
- Contact Bane Sullivan about any troubles using the scripts to build/install plugins
  - Bane Sullivan: [chrsulli@mines.edu](mailto:chrsulli@mines.edu)