Genuage User Manual
Version 1.4.1

September 28, 2021

Abstract

A manual for the Genuage software, a point cloud visualization software utilizing VR technology, mainly intended for use with Single Molecule studies data.
It will detail the various functionalities of the software and how to use them.

Contents

1 Installation and Execution

1.1 System Requirements

SYSTEM REQUIREMENTS (MINIMUM)
GPU : Nvidia Geforce 1050 or equivalent
Operating System : Windows 7, 8 or 10
Other : SteamVR must be installed for VR functionalities to work

SYSTEM REQUIREMENTS (RECOMMENDED)
GPU : Nvidia Geforce 1070 or better
Operating System : Windows 7, 8 or 10
Other : SteamVR must be installed for VR functionalities to work

1.2 Supported VR Hardware

VR Hardware :
- HTC Vive / Valve Index with Vive controllers
- Oculus Rift / Oculus Rift S with Oculus Touch controllers
(Other headsets may work for visualization purposes, but the controller layout was made for HTC Vive controllers.)

1.3 Installation and Launch

The software is packaged into a standalone folder with an executable file, no libraries other than SteamVR are required to install and run it.

2 Desktop Interface Overview

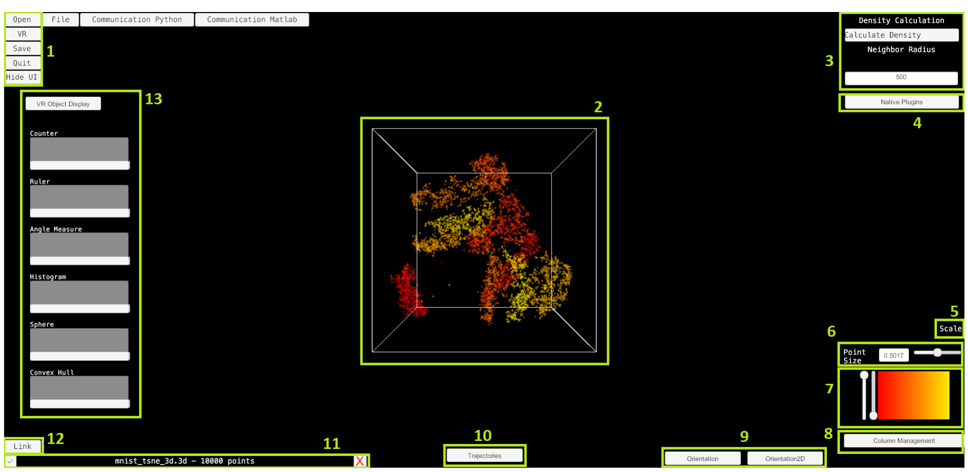

Figure 1: Genuage Desktop Interface

1. Main Menu (Contains buttons to load files, change to VR or Desktop mode, save data, quit the software or hide the on screen UI)
2. Loaded point cloud
3. Density Calculation menu
4. Native Plugins Menu
5. Cloud Point Scaling menu
6. Point Size slider
7. Colormap selection and saturation menu
8. Data Reload menu
9. Orientation Display buttons
10. Trajectory display button
11. Data selection button (click the center part to select, red cross button to close the data, toggle to select the dataset for cloud linking)
12. Cloud Linking button
13. VR Objects Display menu

3 Data Loading

3.1 File Format

Genuage supports ASCII file formats, the conventions are as follow : every row represents one point localization, and columns are separated with tabs, each column represent one parameter of the data: localization, trajectory, orientation...
Genuage can read any text file extension and is unit agnostic.

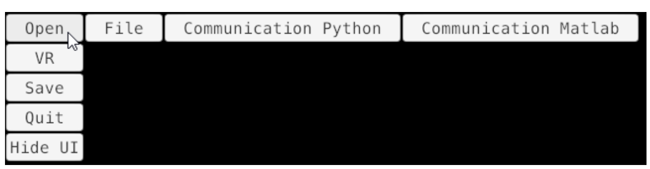

Figure 2: Data Loading UI

3.2 Load Data From File

To load data from a file, click the appropriate button in the load menu, this will open a window where the user can set what separators their file uses, then click the launch browser button to open a browser window to select the files to load.

3.3 Load Data From TCP/IP Transfer (Matlab/Python Supported)

To load data from a byte transfer, click the appropriate button in the load menu, this will open a window informing the user that the software is waiting for the transfer, then, run the script to send the data from your Python or Matlab program of choice to transfer your data. Genuage will load the point after receiving and processing everything.
Scripts for python and matlab are provided.

4 Data Saving

Genuage has functions to save clouds with metadata files so the software can reload the parameters previously set and the operations the user did previously. The software also allows the user to save selected regions of points into a new file.

4.1 Saved Metadata Format

When a full point cloud is saved, Genuage also creates a metadata file in the .JSON format, this file contains the following information • The visual parameters used during the session
• The List of all selected points
• Data about the VR objects placed in the data
• Results of various analyses on the data

Genuage uses these metadata to reload the user's configuration and analyses when the saved file is loaded next.
A matlab script is provided to extract data from the metadata files.

4.2 Save Selected Points (Cropping)

This function saves all the selected points into a new file, no metadata will be saved.

5 Visualization

5.1 Point Cloud Visualization

5.1.1 Point Rendering

The points are rendered as gaussian spheres, with an amplifying effect on their luminosity when they overlap, thanks to a custom shader.

5.1.2 Internal software variables for point clouds

To represent each points, Genuage uses five variables, the position in the x y and z axes, the color value or intensity, and their sizes

5.1.3 Visualization Parameters

ColorMaps

Points are colored according to the normed value of their **Color** variable, applied on a color map texture.
Genuage offers the following standard color maps to colorize the points.

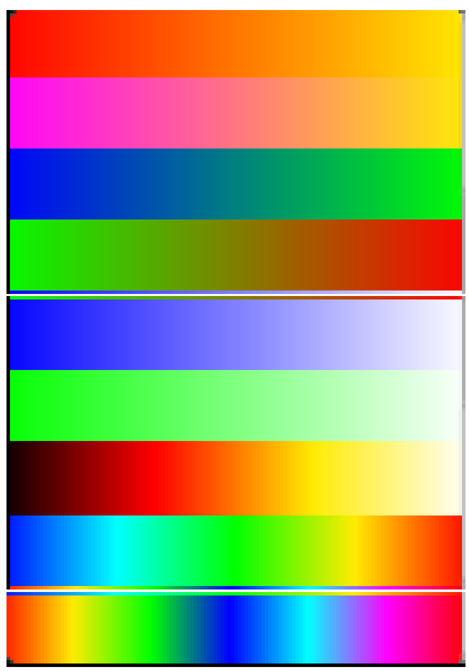

Figure 3: ColorMaps

1. autumn
2. spring

3. winter
4. greenred
5. Blues
6. Greens
7. hot
8. jet
9. hsv

Change Scale

The clouds can be scaled globally with one value or with separate value for each axis.

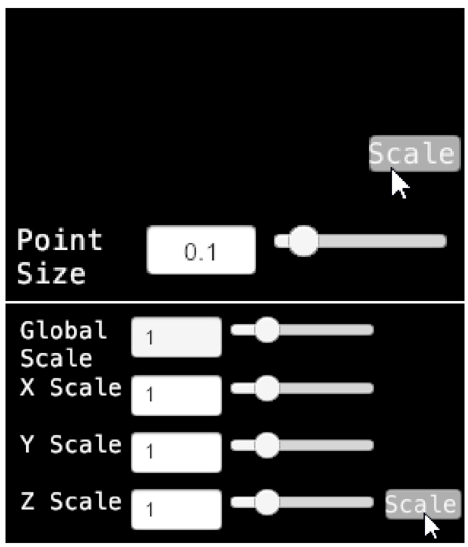

Figure 4: UI used to set the scale.

Change Point Size

The global size of the points can be set, with a slider or a direct input.

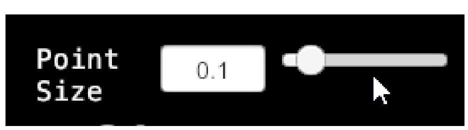

Figure 5: UI used to set the point size

Thresholding

Thresholds can be defined to exclude points above or under specific values for each data column.

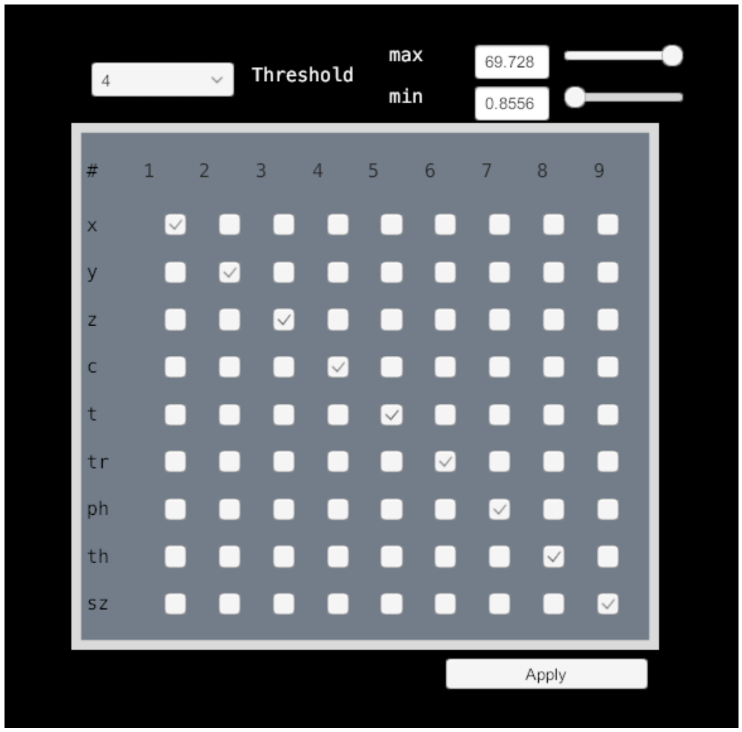

Figure 6: Threshold UI

5.2 Point Trajectories Visualization

5.2.1 Trajectories Rendering

Trajectories are rendered as lines linking the points.
Individual trajectories are represented by the **Trajectory** variable, which is used as the identifier for all the points belonging to one trajectory.
The order of the points in a trajectory are represented by the **Time** variable.
Trajectories are by default colorized by their id on the `Jet` colormap.

5.2.2 Frames Display Options and Animation

the trajectory interface allows the user to visualize the evolution of trajectories in time, either automatically with animations running at different speed, or manually by shifting through the frames of time.

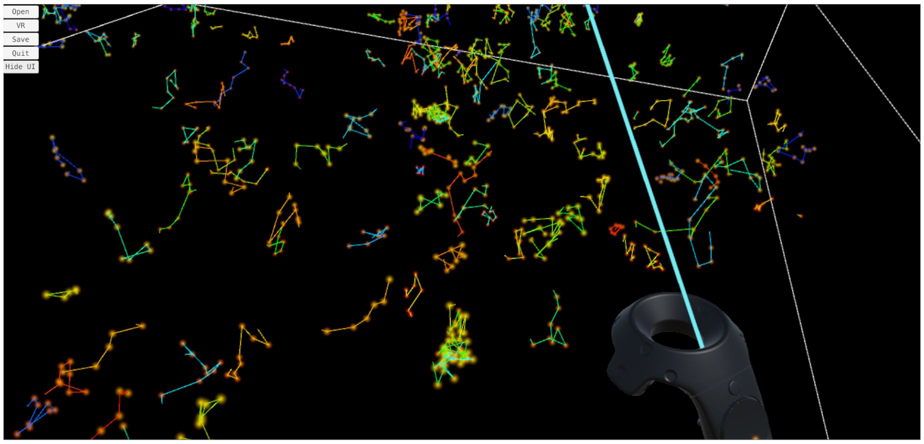

Figure 7: Point Trajectories

The **dragontail** option is used to determine how many points should remain visible after the current frame.

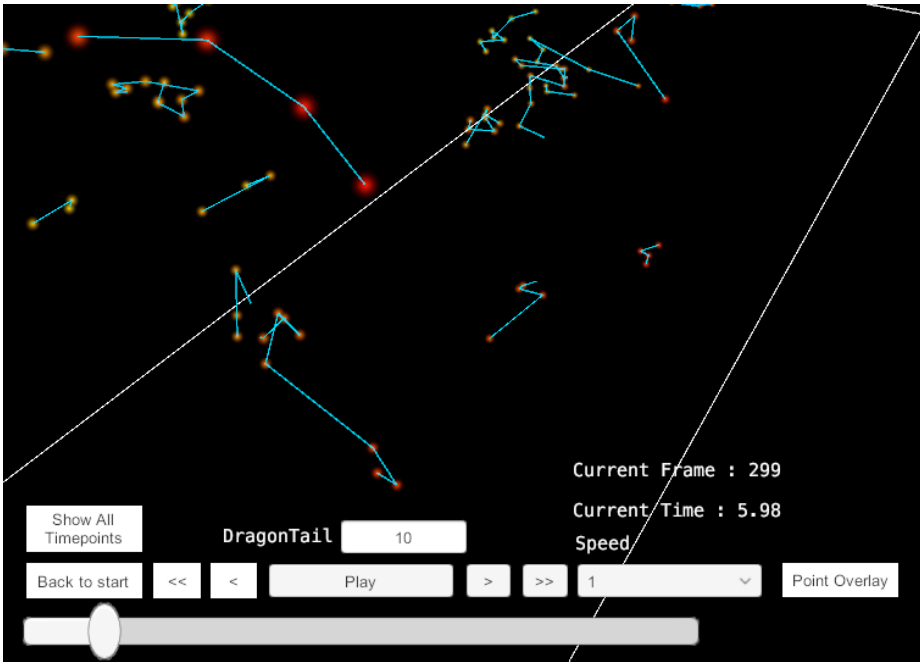

Figure 8: Point Trajectories

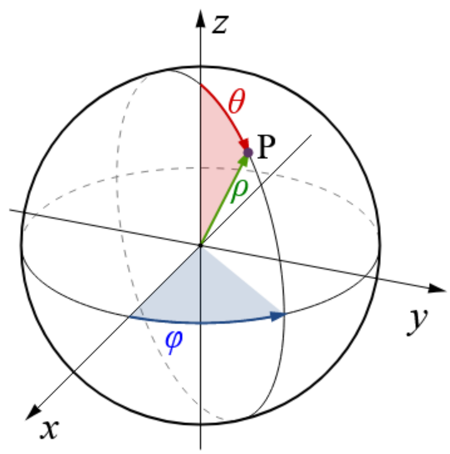

Figure 9: Spherical coordinates system

5.2.3 Internal software variables for trajectories

Trajectories are defined by an id value shared by all the points in the trajectory, the order of apparition of the points is determined by their time value.

5.2.4 Point Overlay

It is possible to overlay the render of the point localizations and even the orientations to the trajectory display.

5.3 Point Orientation Visualization

5.3.1 Orientations Rendering

Orientations are rendered as lines going through the points.

5.3.2 Internal software variables for orientations

Orientations of the points are represented by the two angles of the spherical coordinate system (phi and theta), in radians.

6 Cloud Calculations

6.1 Density Calculation

Genuage can calculate local density for each points of a dataset, with a neighbor radius given by the user.

6.1.1 Density Calculation Method

Because the density calculation method is a high complexity problem, a simplified approach was used in Genuage to speed up processing times when dealing with large amounts of data.

The space occupied by the dataset coordinates is split into a number of regions and all the points each of the regions are detected, when calculating the number of neighbors for each points, the algorithm only looks at all the other points in the region and the adjacent regions.
The local density calculated for each point corresponds to the number of points that have a distance inferior to the specified **Neighbor Radius**.

6.1.2 Density Calculation Result

When the density has been calculated for all points, a new data column appears in the **Data Reload** menu, labelled ”d”.
It represents the number of neighbors within the radius chosen, and can be set to represent any of the **software variables**.
An example of use is to set the **color** variable to this column in order to color all the points based on their density.

7 Operations On Loaded Clouds

7.1 Change software variables configuration and reload the file data

This function is used to change which column represents which software variable for an already loaded dataset. The selection is done by ticking the checkboxes in the reloading window, one column can be linked to several variables at once.

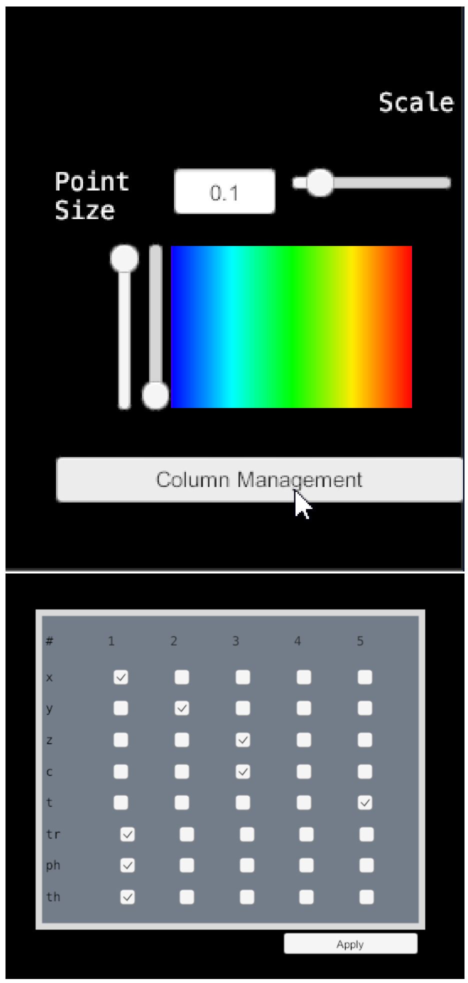

Figure 10:

File Columns		1	2	3	4	5	6	7	8
Software Variables	X	x							
	Y		x						
	Z			x					
	Color				x				
	Time					x			
	Trajectory						x		
	Theta							x	
	Phi								x
	Size								x

7.2 Cloud Linking

The Cloud Link function is used to represent two or more clouds in the same space.

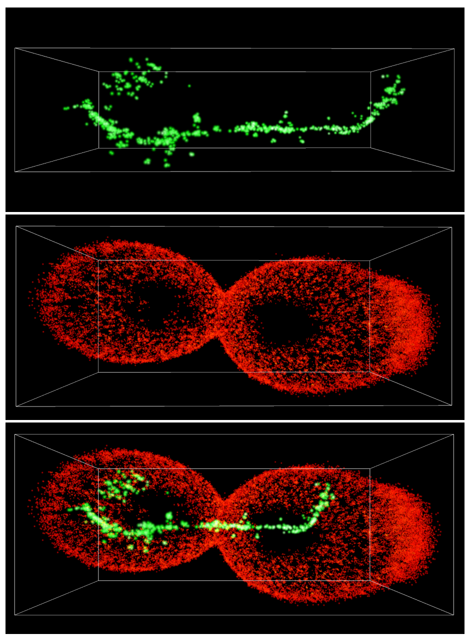

Figure 11: Example of two clouds and the result when they are linked

To link loaded clouds, the toggle at the left side of their selection button must be on.

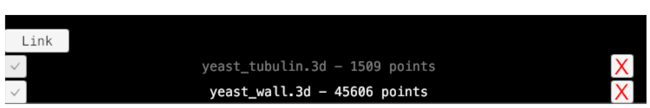

Figure 12:

8 VR Interactions

8.1 Controller interface layout

Below are detailed the button layouts of the controllers supported by the software.
Only one controller is necessary to manipulate the data and use all of the VR tools.

8.1.1 HTC Vive Controller

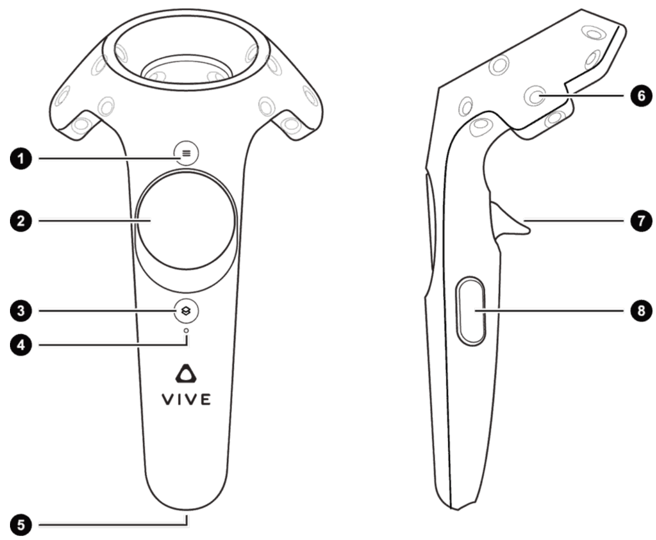

Figure 13: HTC Vive controller button layout

1. VR Tools Menu Button
2. Trackpad (used differently by various tools, see further sections)
3. Controller Power Button
4. Power Light
5. Micro USB Charging Port
6. IR Receptors for tracking
7. Trigger (used to grab objects and interact with the VR interface)
8. Side Grips (not used)

8.1.2 Oculus Touch Controller

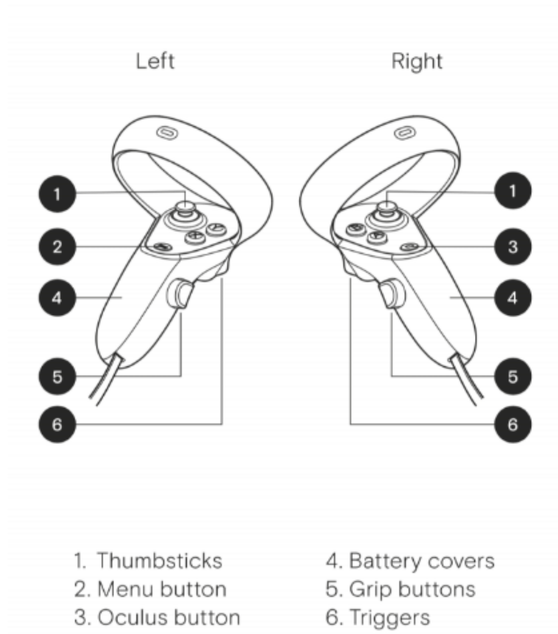

Figure 14: Oculus-touch-controllers-1

1. Thumbstick (Used like the HTC VIVE trackpad for various tools)
2. Reserved button
3. Reserved button
4. Battery cover
5. Side grip (unused)
6. Trigger (used to grab objects and interact with the VR interface)
7. IR Receptors for tracking

9 VR Menu

The VR menu allows the user to activate various VR tools and to use specific features of Genuage's desktop mode such as setting visual parameters or select loaded clouds.

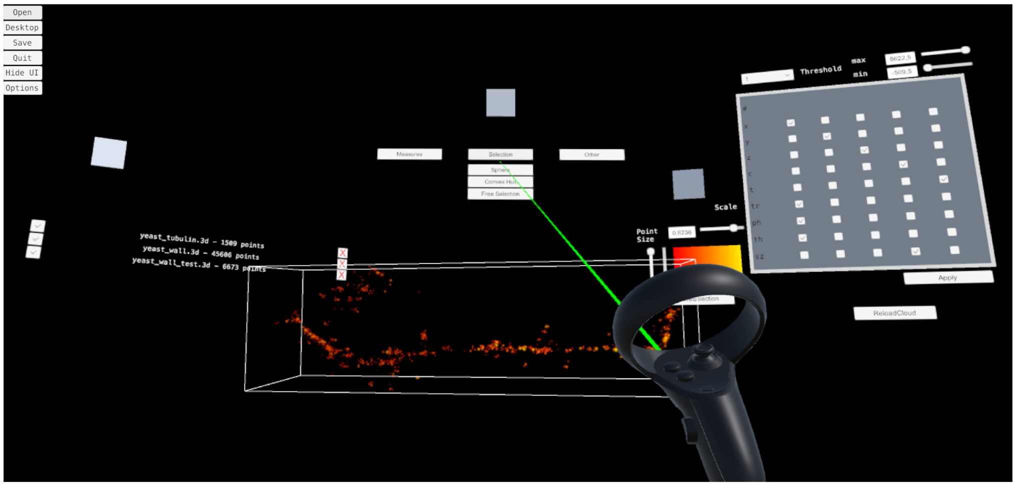

Figure 15: VR Menu

9.1 Tool Menu

The tool menu allows the user to activate the various VR tools.

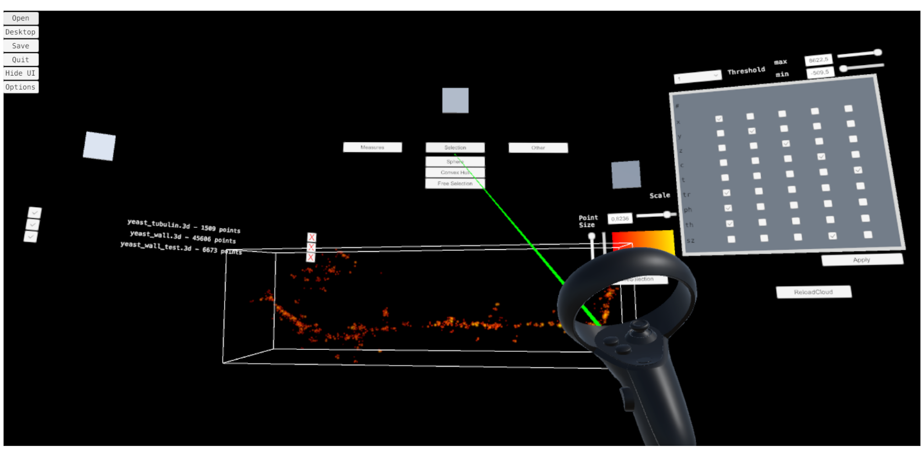

Figure 16: VR Menu

9.2 Visualization Parameters and Cloud Selection

The Visual parameters menu allows the user to modify the color map and color saturation, point size and scale of the active cloud, as well as reconfigure the software variables. The user can also save the currently selected points to a new point cloud.
The cloud selection menu allows the user to change the selected cloud and delete loaded clouds.

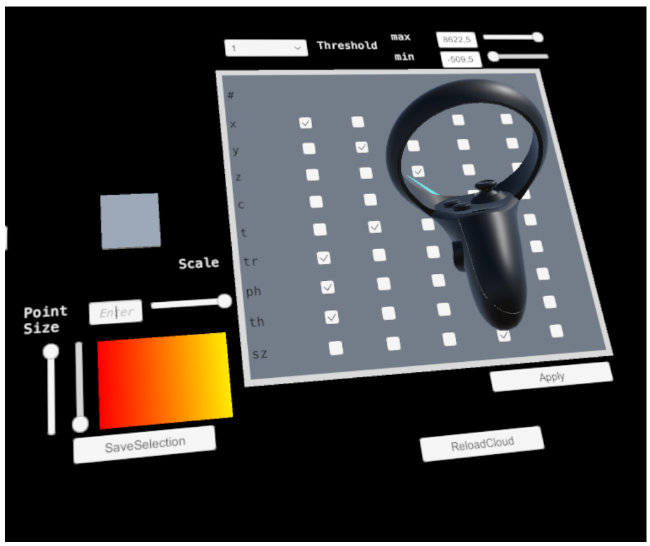

Figure 17: VR Visual Parameters Menu

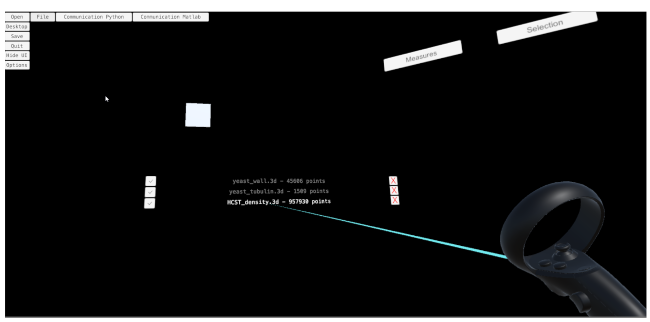

Figure 18: VR Cloud Selection Menu

10 VR Tools

10.1 Pointer Tool

The Pointer tool allows the user to grab some objects and interact with vr interface elements by pointing the controller at them and pressing the **Trigger**.

The laser pointer linked to the controller model is used to represent the tool, it is blue normally and turns green when it points to an object that can be interacted with.
The Pointer tool is always active.

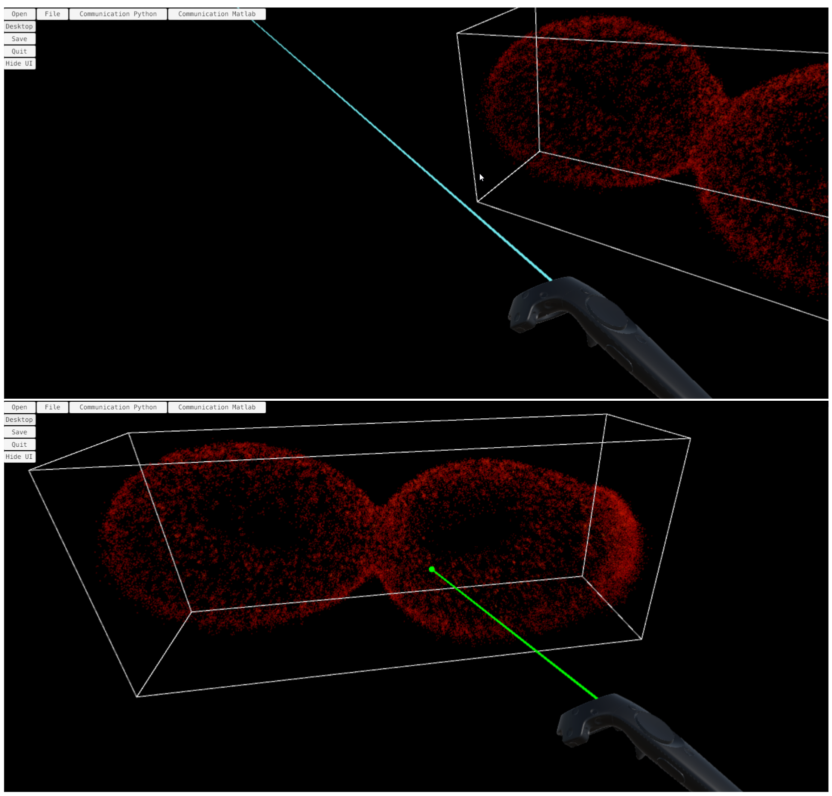

Figure 19: Pointer tool

10.2 Interface for the Object Placement Tools

The Object Placement Tools are tools allowing the user to place objects inside the data for various purposes. All these tools use the same interface for object placement.

The objects placed all belong to an Object Set, and multiple sets can be created. To do this the **Trackpad** of the controller is used.
The **Trackpad** is a tactile part of the HTC Vive controller, here different zones of the trackpad will correspond to different buttons. An image is associated with each zones, and when the user puts their finger on the corresponding side, the image is colored in green and a descriptive text pops up next to it to explain what the button does.

The layout of the two interfaces, one to place the objects, the other to select the sets, are detailed below :

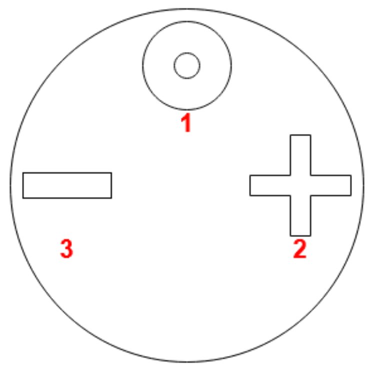

Figure 20: interface for object placement

1. Go to Set Selection (activates the set selection interface)
2. Add a new object to the set
3. Delete an object from the current set (said object must be touched by the sphere pointer)

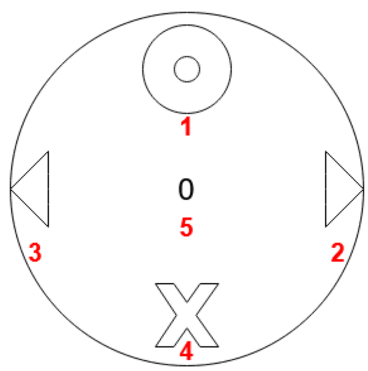

Figure 21: interface for object set selection

1. Go to Object Placement for the current set(activates the object placement interface)
2. Go to next set
3. Go to previous set
4. Delete current set
5. Current set

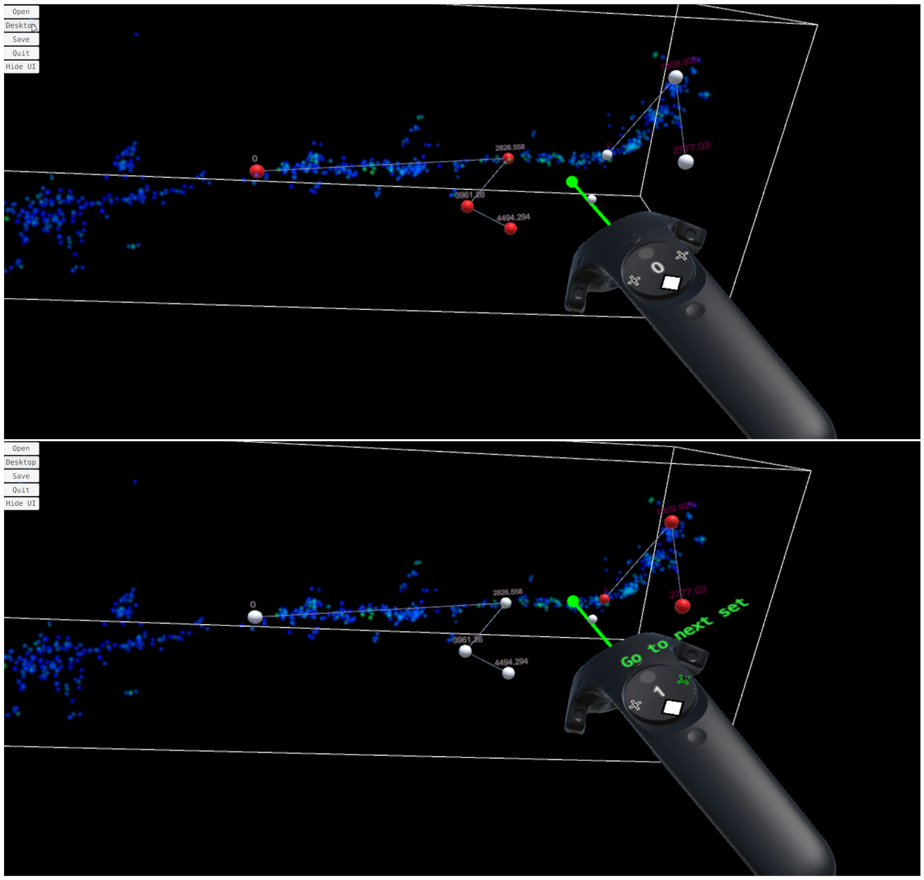

Figure 22: Example of the Set system

Here the points making up the two distance measures belong to Set 0 and 1 respectively and they are highlighted in red when the corresponding set is selected with the Set Selection Interface.

10.3 Object Placement Tools - Quantification Tools

10.3.1 Counter Tool

The **Counter** tool is a simple tool used to place numbered spheres inside the data. Each sphere set has a specific color for the text.

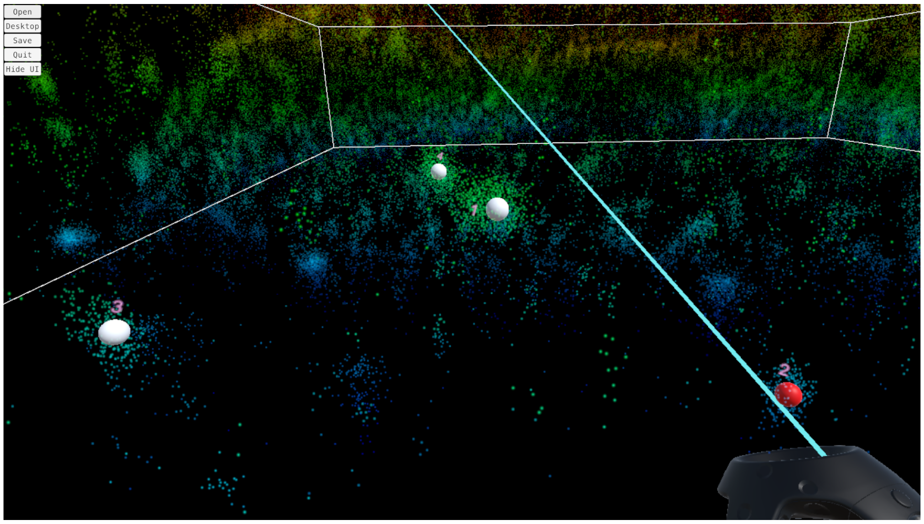

Figure 23: Counter Tool

10.3.2 Ruler Tool

The **Ruler** tool is used to measure distances inside the data. The spheres placed are linked and the distance from each sphere to the first one is displayed above it. Each sphere set represent one measure.

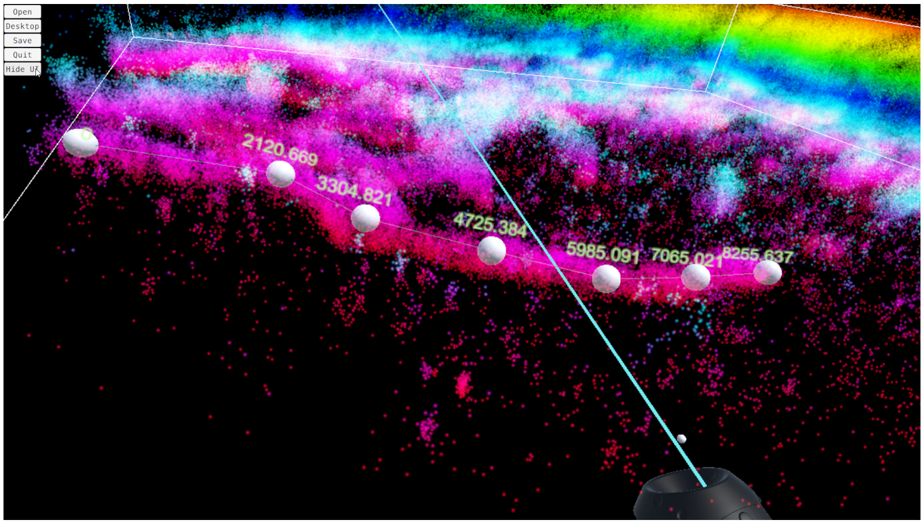

Figure 24: Ruler Tool

10.3.3 Angle Measure Tool

The **Angle Measure** tool is used to measure angles inside the data. Every two spheres placed form a segment, and once four sphere have been placed, the angle in degrees between the two segments is displayed. Each set represents one angle.

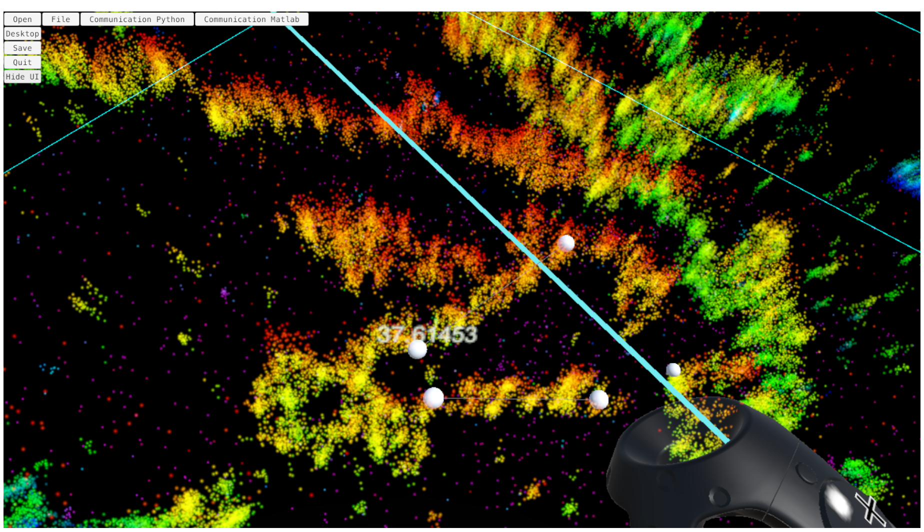

Figure 25: Angle Measure

10.3.4 Profiler Tool

The **Profiler** tool is used to get a profile of the point distribution inside an area of the dataset.
The tool places a cylinder inside the data to determine the area to measure into, then divides the cylinder into multiple slices or blins.
All the points in each blin will be counted and the results displayed as a histogram in a VR window.

To create the cylinder, push the **Add Object** button a first time, this will create both ends of the cylinder, the first one staying at the place the controller was pointing to when the button was pushed, and the other will stick to the controller, allowing the user to move it.
Then place the controller to where the other end of the cylinder should be. Press the **Add Object** button a second time to validate the position.
After both ends of the cylinder are placed, move the controller closer or away from it to set the height of the cylinder, press the **Add Object** button a second time to validate the height.
When the heigh is set, the cylinder is complete, and the controller will switch to the blin selection interface. When the number of blins in the cylinder is changed, a result VR window will appear and display a histogram of the number of points in each section of the cylinder.

When the operation is complete, use the **Validation button** of the interface to finish the process and return to the **Object Placement interface**.

The interface for the blins is detailed below :

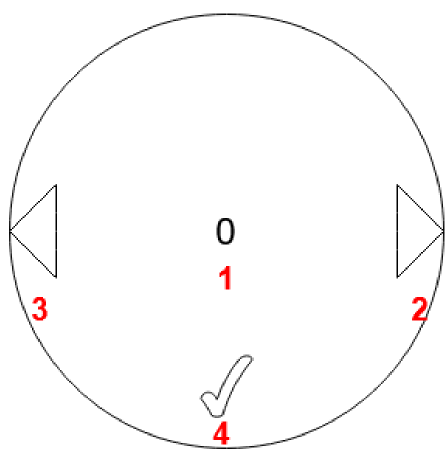

Figure 26:

1. Number of blins
2. Add one more blin
3. Substract one blin

4. Validate the number of blins and exit to Object Placement Interface

10.3.5 Object Placement Tools - Selection Tools

10.4 Note on Point Selection

To determine the points that are selected, Genuage sums up all the sources of selection (I.E all the selection regions created by the tools detailed below).
Furthermore, selected points that are marked as hidden (points that are outside the thresholding limits for example) will not be taken into account for any selection operation.

10.5 Selection Sphere Tool

The **Selection Sphere** tool allows the user to draw a sphere inside the data. To use it, place the controller where the center of the sphere should be, press and hold the **Add Object** button. While holding the button, move the controller away to set the preferred radius for the sphere, once it's set, release the button to end the placement.
Once the sphere is set, all the points inside will be marked as selected, their color turning green.
Each set represents one sphere.

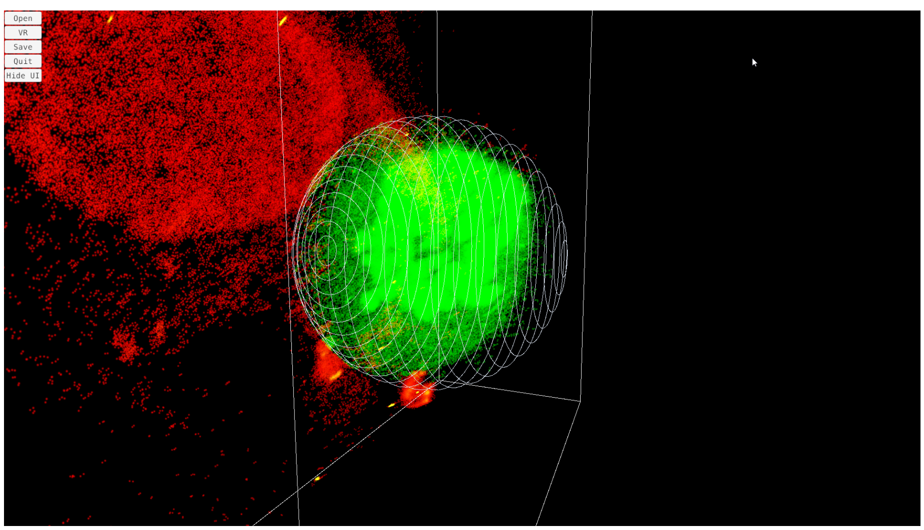

Figure 27: Sphere

10.5.1 Selection Convex Hull Tool

The **Selection Convex Hull** tool is used to draw a custom shape to select specific areas. It places spheres inside the data, once four spheres have been placed, they are linked and form a volume. All the points

inside are marked as selected and colored green. Further points added will expand and shape the volume and select more points.
Each set represents one convex hull and one selection volume.

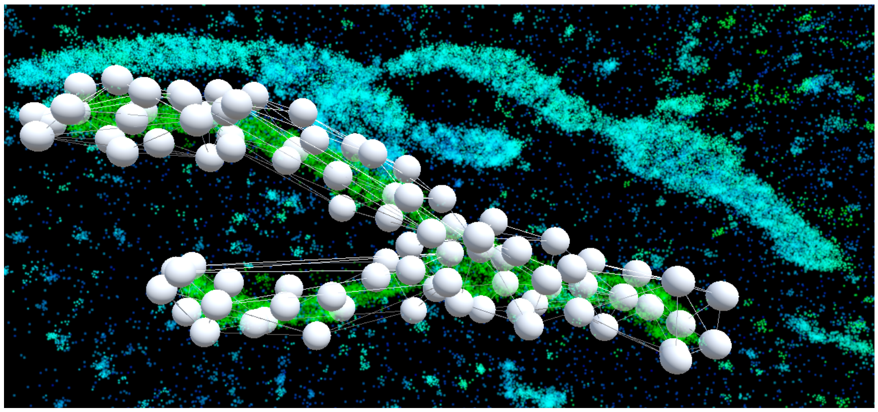

Figure 28: Convex Hull

10.6 Other Selection Tools

10.6.1 Free Selection Tool

The free selection tool allows the user to select simple structures. A transparent sphere represents the selection area. Points inside the sphere are highlighted in white. The user can set the size of the sphere with the side buttons, select every point in the sphere with the up button or delete every point in the sphere from the selection with the down button.

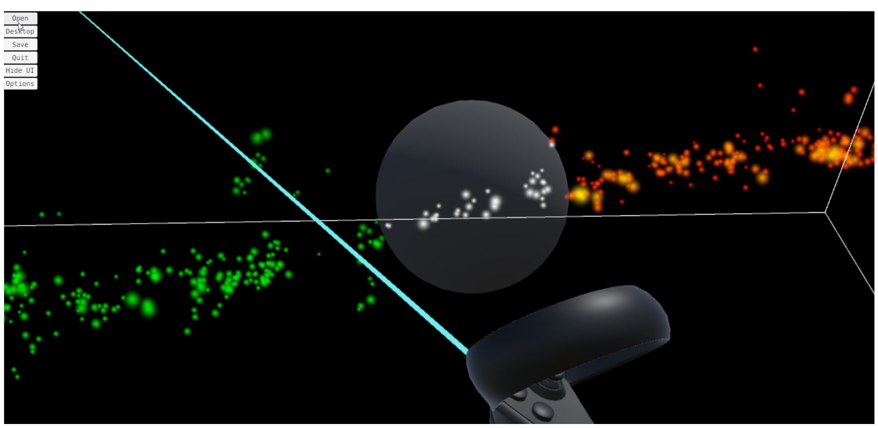

Figure 29: Free Selection Tool

10.7 Visualization Tools

10.7.1 Clipping Plane Tool

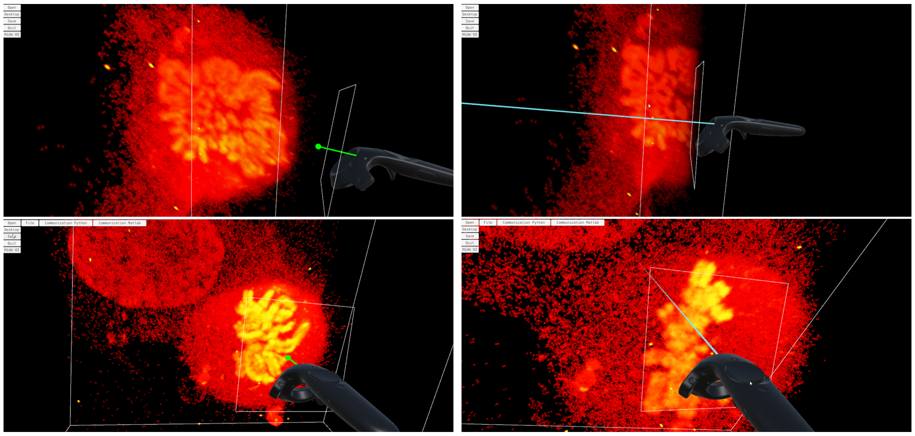

Figure 30: Cliping Plane

The **Clipping Plane** tool puts a plane, represented by a square in front of the controller that will remove all points behind it from display. It allows the user to dig into the data using the controller.
This tool is compatible with the Object Placement Tools.

10.7.2 Multi Clipping Plane Tool

The multi clipping plane tool allows the user to set fixed clipping planes in a cloud. Up to 10 planes can be set at the same time.

11 Native Plugins : C/C++ powered analysis tools

Native Plugins are libraries of C or C++ code that can be accessed by Genuage to perform various analyses.

Below is a list of the pre-installed native plugins and their functions.

11.1 Infer3D : 3D Inference on trajectories

Infer3D is a Bayesian inference script used to calculate the diffusion coefficient and possibly the 3D force of a sample of point trajectories.
Infer3D runs on a sample of point trajectories that are taken from all the selected areas in the currently active point cloud.

The INfer3D code optimizes the result a given function that can be selected in the UI :

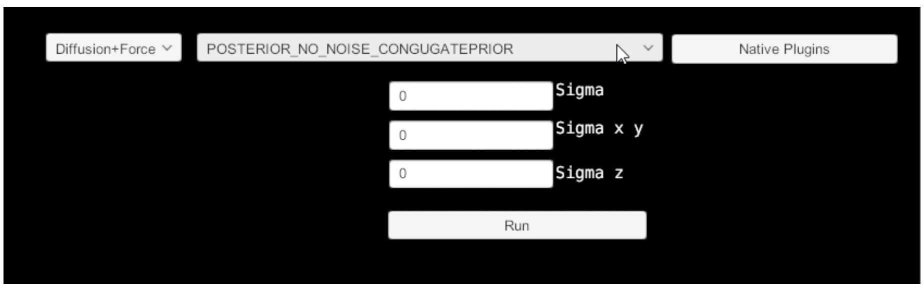

Figure 31: UI for the Infer3D plugin

The different functions that can be selected represent different hypotheses on the environment, diffrenent priors and porteriors.
Three variables can also be set, **Sigma**, **Sigmaxy** and **Sigmaz**, they represent the noise in microns, either the global noise for **Sigma**, the noise on the x and y axes for **Sigmaxy** and the noise on the z axes for **Sigmaz**

List of all the function options

Pure diffusion functions :

1. Likelyhood No noise
2. Likelyhood Uniform noise *
3. Likelyhood Asymetric noise **/***
4. Likelyhood noise only in z ***
5. posterior no noise non informative prior
6. posterior no noise conjugate prior
7. posterior uniform noise non informative prior *
8. posterior asymetric noise non informative prior **/***
9. posterior noise only in z non informative prior ***

Diffusion and velocity functions :

1. simple Posterior function *
2. Likelyhood No noise
3. Likelyhood Uniform noise *
4. Likelyhood Asymetric noise **/***
5. Likelyhood noise only in z ***
6. posterior no noise non informative prior
7. posterior no noise conjugate prior
8. posterior uniform noise non informative prior *
9. posterior asymetric noise non informative prior **/***
10. posterior noise only in z non informative prior ***

** : uses the Sigma variable*
*** : uses the Sigma x y variable*
**** : uses the Sigma z variable*

If the velocity has been calculated, an arrow will be placed in the center of the selected area to represent the force

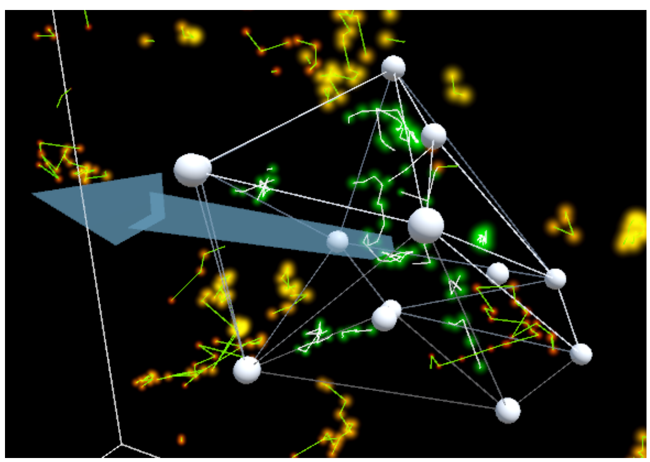

Figure 32: Velocity arrow

11.2 3D Diffusion Map Generation

This feature allows for the generation of 3D maps of the diffusion properties of dynamic data.

The 3D maps are generated in a 3-step process.

First, a K-means Clustering algorithm is used to partition the points into a number of clusters based on proximity. The K-means algorithm iterates over a predefined set of starting clusters. Users can set the number of clusters or partitions, as well as the maximum number of iterations for the algorithm to perform. A new data column is added to the cloud at this step, each entry being the id number of the cluster the point belongs to.

Second, Bayesian inference is performed on each cluster of points, using the Infer3D algorithm described previously. A diffusion coefficient is thus calculated for all the dynamic points within each cluster. User can set various parameters for the priors and the level of noise to be considered by the algorithm. Similarly, a diffusion force vector can be estimated. A new data column is later added to the cloud containing the corresponding diffusion coefficient.

Finally, an implementation of the Quickhull algorithm is used to create a 3-dimensional convex hull mesh for the points in each cluster. The resulting meshes can be displayed as solids, colored according to the values of the diffusion coefficient mapped to one of Genuage's colormaps, or as wireframes. The visibility and opacity of rendered meshes and surfaces can be controlled by the user. The force vectors for the diffusion within each cluster, if they are calculated are displayed as an arrow with a size proportional to the force vector's norm.

The meshes for the polygons generated by the 3D physical map module are generated with split vertices, so that no two triangles share the same vertex. This allows for the edges of each face of the polygons to be lit separately by Unity's lighting engine and for the edges of the solids to be clearly visible when rendered. The meshes are rendered with a modified version of Unity's standard surface shader, in order to take advantage of Unity's lighting system and be compatible with Genuage's shader-based interactions like the VR Clipping Plane tool.

12 Loading and executing user-made DLLs in Genuage

Users can now use their own analysis algorithms in Genuage by importing custom made DLLs. Currently, 3 base functions are implemented in Genuage, Dlls can add new columns to a loaded dataset, calculate values and display them, and send back column data to create a new cloud. Below is a list of the DLLs implemented by the Genuage team and furnished with the software.

12.1 Create new data column : Density DLL

The purpose of this DLL is to estimate for each point the number of the neighbors within a given 3D radius. The user inputs the desired radius. The DLL calculates the number of neighbors for each points within the radius and returns a new data column.

12.2 Calculate a value : Apparent Diffusion DLL

The DLL is dedicated for dynamic point cloud recordings that are generated from single particle tracking experiments. The diffusion of the particles is highly dependent on its physical characteristics (size and functionalization) as well as the surrounding media. As a first order approximation, apparent diffusion coefficient of each trajectory is retrieved from the mean-square displacement (MSD) at different time steps (Normanno et al., 2015; Nora et al., 2020). The apparent diffusion coefficient is estimated from the MSD values that are computed between 2dt and 5 dt, dt being the exposure time of each acquisition frame. The DLL estimates the diffusion coefficient D of a selected trajectory by a linear fitting of the MSD. The slope is related to the apparent diffusion coefficient by: slope=$2\times N\times D$. Here N stands for the dimensionality of the data, e.g. 2 for 2D and 3 for 3D recordings. The DLL returns the estimated diffusion coefficient of the selected trajectory based on the dimensionality of the data.

12.3 Create a new cloud : Correct Blinking DLL

Analyzing single-molecule localization data sets requires specific considerations especially for super-resolution microscopy data sets. A common artifact arises from the blinking of single molecules. The same molecule can last on multiple frames and thus appears as multiple detected points. The reconstructed image can thus conceal the real underlying molecular distribution. Points arising from the same molecule need to accounted for and combined into single detection.

This DLL is dedicated for the correction of multiple appearances due to blinking of the same molecule in PALM and STORM imaging. The DLL combines the localizations that are found at a given spatial and temporal proximity. The exact temporal and spatial extents can be estimated by several establishefind methods (Betzig et al., 2006; Rust et al., 2006; Annibale et al., 2011; Coltharp et al., 2012; Puchner et al., 2013; Bohrer et al., 2021). The values are entered manually as numerical input when executing the DLL.

The code loops over all the frames of the input point-cloud dataset. For each frame, a second loop is performed over a number of frames within search time window (e.g. over 20 frames if the search time is defined at 1s and the acquisition frame rate is 50 ms). The distances between all the molecules of the considered frame and all the molecules of the following frames are computed. Localizations that can be considered as coming from a same blinking molecule are identified and labelled with an identical index. At the end, the algorithm returns for each localization of each frame its index corresponding to a given unique molecule. Based on this information, a mean position and a mean frame number is computed for each multiple-appearing molecule. This DLL generates finally a new point cloud with a different number of elements compared to the original one.