

Supplementary for bioWeb3D: an online webGL 3D data visualisation tool

Jean-Baptiste Pettit* and John C. Marioni

EMBL-EBI, European Molecular Biology Laboratory - European Bioinformatics Institute,
Cambridge, CB10 1SD, UK

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXXX

1 INPUT FILE FORMATS

1.1 Dataset files

When the user adds a new *Dataset* file, a new Dataset section is created in the “Data” panel of the application. One raw data file contains one dataset. The dataset is composed only of 3D coordinates (x,y,z) along with a small amount of additional (e.g., an optional name, an optional “chain” that has to be set to true to link the points together). Below is an example of a minimal 3 point dataset file:

```
1 { "dataset" : {  
2   "name" : "my superb dataset",  
3   "chain" : true ,  
4   "points" : [  
5     [  
6       0.5 ,  
7       100 ,  
8       -50.5  
9     ],  
10    [  
11      200 ,  
12      10 ,  
13      0.0  
14    ],  
15    [  
16      3 ,  
17      250.15 ,  
18      15  
19    ]  
20  ]  
21 }  
22 }
```

1.2 Information layer files

The *Information layer* file contains information about the points described in the Dataset file. The information entered in this file has to be inputted in the same order as the points defined in the Dataset file. Multiple information layers can be defined in the same file as follows:

- a name
- a number of categories called numClust
- A list of labels for the clusters (optional)

For example coming back to the 3 points defined previously, two information layers could correspond to:

- one clustering algorithm that puts the first two points together in the cluster one and the third point alone in a second cluster
- a second clustering algorithm that puts each point in a separate cluster

In this case the Information layer file would look like:

```
1 { "cluster" :  
2   [  
3     {  
4       "name": "clustering algo 1",  
5       "numClust": "2",  
6       "labels" : [  
7         "Category 1",  
8         "Category 2"  
9       ],  
10      "values": [  
11        1 ,  
12        1 ,  
13        2  
14      ]  
15    },  
16    {  
17      "name": "clustering algo 2",  
18      "numClust": "3",  
19      "values": [  
20        1 ,  
21        2 ,  
22        3  
23      ]  
24    }  
25  ]  
26 }
```

2 CONVERTING CSV FILES TO JSON

A lot of data generated in biology is stored within CSV files. Converting CSV document to the Json format used in this application can be a bit complicated. We provide scripts written in Perl to do this conversion. Of course to work the converters need the CSV files to follow a certain format.

2.1 CSV to Dataset file

To use the “csv_to_dataset.pl” converter, the input CSV file must contain only the points coordinates. Each line representing a point, the three coordinates on each line must be separated by “tabulation” characters. Example :

```
1 0.5 100 -50.5
2 200 10 0.0
3 3 250.15 15
```

The syntax to use the converter is the following :

```
1 perl csv\_to\_dataset.pl [csv file] [dataset name]
   [Chain parameter : true if the points should
   be linked, false otherwise]\\
```

for example if, the previous CSV file is named “example.csv”, and the points should be linked, then the command line will be:

```
1 perl csv\_to\_dataset.pl example.csv my\_dataset
   true
```

The result file named “example.csv.json” will contain :

```
1 { "dataset" : {
2   "name" : "my_dataset",
3   "chain" : true,
4   "points" : [
5     [
6       0.5,
7       100,
8       -50.5
9     ],
10    [
11      200,
12      10,
13      0.0
14    ],
15    [
16      3,
17      250.15,
18      15
19    ]
20  ]
21 }
22 }
```

2.2 CSV to information layer file

To use the “csv_to_information_layer.pl” converter, each column of the CSV files will be one information layer. The first element of each column will be the name of the information layer the rest of the column represents in which class each point belongs. The separation character between columns must be a “tabulation”. for example :

```
1 clustering\_algo\_1 clustering\_algo\_2
2 1 1
3 1 2
4 2 3
```

The syntax to use the converter is the following :

```
1 perl csv\_to\_information\_layer.pl [csv file]
```

for example if, the previous CSV file is named “example_information_layer.csv”, and the points should be linked, then the command line will be:

```
1 perl csv\_to\_information\_layer.pl example\_
   information\_layer.csv
```

The result file named “example_information_layer.csv.json” will contain :

```
1 { "cluster" :
2   [
3     {
4       "name": "clustering_algo_1",
5       "numClust": "2",
6       "values": [
7         1,
8         1,
9         2
10      ]
11    },
12    {
13      "name": "clustering_algo_2",
14      "numClust": "3",
15      "values": [
16        1,
17        2,
18        3
19      ]
20    }
21  ]
22 }
```

Please note that at the moment it is not possible to use the “labels” property with this converter.