BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–3

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: XXXXXXX

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 information (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:

```
{ "dataset" : {
      "name" : "my superb dataset",
      "chain" : true,
       points" :
        [
          [
            0.5,
            100,
            -50.5
          ],
          [
            200,
            10,
            0.0
          ],
          [
            3,
            250.15,
            15
          ]
        ]
     }
}
```

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 numClass
- A list of labels for the classes (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 classe one and the third point alone in a second class
- a second clustering algorithm that puts each point in a separate class

In this case the Information layer file would look like:

```
{ "information" :
  [
    {
      "name": "clustering algo 1",
      "numClass": "2",
      "labels" : [
        "Category 1",
        "Category 2"
      ],
      "values": [
        1,
        1,
        2
      ]
    },
    {
      "name": "clustering algo 2",
      "numClass": "3",
      "values": [
        1,
        2,
        3
      ]
    }
  ]
}
```

2 CONVERTING CSV FILES TO JSON

Much data generated in the biological sciences is stored within CSV files. Converting CSV document to the JSON format used in this application is not always trivial. In order to facilitate this process, we provide scripts written in Perl to perform the conversion. We describe hereafter the CSV formats handled by the converters.

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 represents a point and the three coordinates on each line must be separated by "tabulation" characters. Example :

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

The syntax to use the converter is the following :

```
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:

```
perl csv_to_dataset.pl example.csv
    my_dataset true
```

The result file named "example.csv.json" will contain :

```
{ "dataset" : {
      "name" : "my_dataset",
      "chain" : true,
        "points" : [
          [
            0.5,
            100,
            -50.5
          ],
          [
            200,
            10,
            0.0
          ],
          [
            3,
            250.15,
            15
          ]
        ]
    }
}
```

2.2 CSV to information layer file

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

```
clustering_algo_1 clustering_algo_2
1 1
1 2
2 3
```

The syntax to use the converter is the following :

```
perl csv_to_information_layer.pl [csv file]
```

For example if, the previous CSV file is named "example_information_layer.csv", then the command line will be:

```
perl csv_to_information_layer.pl
    example_information_layer.csv
```

The result file named "example_information_layer.csv.json" will contain :

```
{ "information" :
  [
    {
      "name": "clustering_algo_1",
      "numClass": "2",
      "values": [
        1,
        1,
        2
      ]
    },
    {
      "name": "clustering_algo_2",
      "numClass": "3",
      "values": [
        1,
        2,
        3
      ]
    }
  ]
}
```

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

3 FURTHER EXAMPLES

3.1 Visualizing simultaneous worlds

You can choose to split the application screen in up to 4 different worlds in order to visualize and compare different information layers of the same dataset or different datasets (figure 1).

3.2 Visualizing sequential information

If the "chain" property in the dataset file is set to "true", the point will be linked together in the 3D visualization (figure 2).

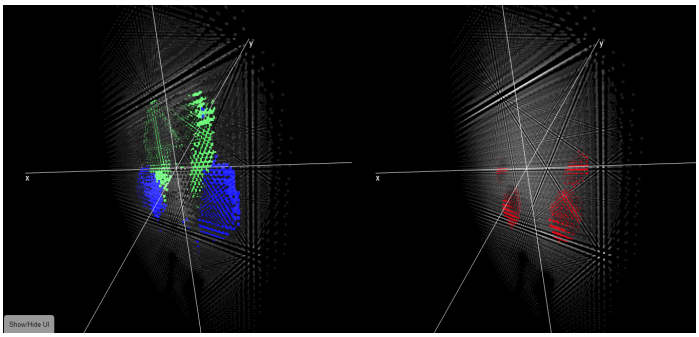

Fig. 1. In this figure, the left-hand world shows two clusters (in blue and green) in the brain of the marine annelid *Platynereis dumerilii* and the right-hand world shows the expression data for the gene C-Opsin (in red) in the same dataset. Data for this figure was taken from (Tomer *et al.*, 2010).

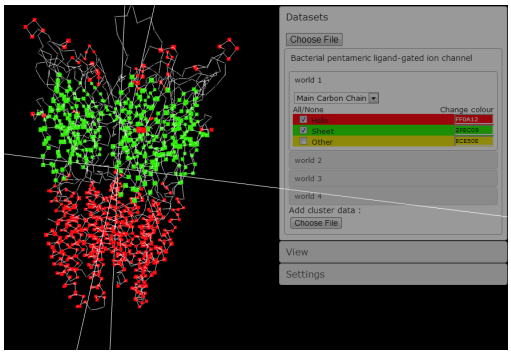

Fig. 2. An example of a sequential dataset where the points are linked together. Here we present the 3D structure of the main carbon chain of a bacterial pentameric ligand-gated ion channel. In red we show the carbon atoms which are part of an α-helix secondary structure, in green we show the atoms which are part of a β-sheet secondary structure. Data for this figure was taken from the Protein Data Bank in Europe website

REFERENCES

Tomer R., Denes A.S., Tessmar-Raible K., Arendt D. (2010). Profiling by image registration reveals common origin of annelid mushroom bodies and vertebrate pallium. *Cell*, **142**:800-809.