Json format

Json file containing:

```
{
   <soundInfo>,
   <dataSet>,
   <processingMethod>,
   <colorBy>,
   <totalPoints>,
   <tags>,
   <points>
}
```

All fields are required

<soundInfo>

```
"soundInfo": <null | str>
```

Where the value is either null if there is no reference data to link to; or a path to a .json file containing reference data.

<dataSet>

```
"dataSet": <str>
```

Where the value is the name of the dataset used. E.g. “`nsynth`”.

<processingMethod>

```
"processingMethod": <str>
```

Where the value is a description of the specific map to differentiate it from others using the same dataset.

<colorBy>

```
"colorBy": <str>
```

Where the value is the name of the tag to use for determining the default coloring of points.

<totalPoints>

```
"totalPoints": <int>
```

Where the value is the total number of points in the map.

<tags>

```
"tags": {
    <tag>,
    ...
    ...
    <tag>
}
```

Contains an arbitrary number of tag objects.

<points>

```
"points": [
    <point>,
    ...
    ...
    <point>
]
```

Where the value is an array containing exactly the number of `<point>` objects specified in `<totalPoints>`.

<tag>

```
"<tag_name>": {
    <__filterable>,
    <value>,
    ...
    ...
    <value>
}
```

Where the key is the name of the tag. Contains at least one `<value>` object and 0 or 1 `<__filterable>` objects. The order of `<value>` objects determines the order of values when filtering.

<point>

```
[<float>, <float>, <float>, <str>]
```

Where the x, y and z coordinates of the object are in order, followed by the name of the audio file. If the projection is 2-dimensional, the value of the third float is 0.

<value>

```
"<tag_value>": {
    <color>,
    <points_t>
}
```

Where the key is the value of the tag. E.g. for phoneme tags the value could be “a”.

<__filterable>

```
"__filterable": <bool>
```

Where the value is either `false` or `true`, depending on whether the tag in question should be filterable. `__filterable` is a reserved keyword and can not be used as a tag value.

<color>

```
"color": <html_hex>
```

Where the value is an html compatible hexadecimal representation of the color. E.g. `#FF0000` or `#FFFFFF`.

<point_t>

```
"points": [<id1>, … , <idn>]
```

Where the value is a table containing at least one point id. I.E. a list of points that have the value in question.

Below is an example json file of phoneme data:

```
{
   "soundInfo":null,
   "dataSet":"phoneme",
   "processingMethod":"ms - t-SNE, 30",
    "colorBy":"phoneme",
    "totalPoints":13,
    "points":[
        [167.9238936178313, 254.04705572486748, 0, "mv_0693_003_h_0_0.wav"],
        [352.73155797694244, 300.65700235519427, 0, "mv_0693_001_k_0_0.wav"],
        [222.59116734148247, 455.1318404632827, 0, "mv_0693_002_a_1_0.wav"],
        [287.32101980267936, 600.0, 0, "mv_0693_008_C331_1_0.wav"],
        [410.3317393631231, 449.93148927931816, 0, "mv_0693_005_i_1_0.wav"],
        [236.7373891954687, 175.78310947738692, 0, "mv_0693_007_o_1_0.wav"],
        [303.6777517369889, 322.19023618154307, 0, "mv_0693_001_k_0_1.wav"],
        [247.8584758639965, 257.86051342074495, 0, "mv_0693_004_v_1_0.wav"],
        [572.6179909435875, 531.9914175063502, 0, "mv_0693_003_h_0_1.wav"],
        [316.43828789883827, 411.6092810029525, 0, "mv_0693_006_a_1_0.wav"],
        [364.59864125072744, 0.0, 0, "mv_0693_008_C331_1_1.wav"],
        [441.61125865209823, 214.40766642376047, 0, "mv_0693_002_a_1_1.wav"],
        [428.6061042336219, 368.26174292319473, 0, "mv_0693_007_o_1_1.wav"]
    ],
    "tags":{
        "file name":{
            "__filterable":false,
            "mv_0693_005_i_1_0.wav":{
                "points":[4],
                "color":"#0091ff"
            },
            "mv_0693_007_o_1_1.wav":{
                "points":[12],
                "color":"#b600ff"
            },
            "mv_0693_008_C331_1_0.wav":{
                "points":[3],
                "color":"#ff00da"
            },
            "mv_0693_001_k_0_1.wav":{
                "points":[6],
                "color":"#ffda00"
            },
            "mv_0693_002_a_1_0.wav":{
                "points":[2],
                "color":"#b6ff00"
            },
            "mv_0693_006_a_1_0.wav":{
                "points":[9],
                "color":"#0024ff"
            },
            "mv_0693_003_h_0_1.wav":{
                "points":[8],
                "color":"#00ff91"
            },
            "mv_0693_003_h_0_0.wav":{
                "points":[0],
                "color":"#00ff24"
            },
            "mv_0693_001_k_0_0.wav":{
                "points":[1],
                "color":"#ff6d00"
            },
            "mv_0693_004_v_1_0.wav":{
                "points":[7],
                "color":"#00ffff"
            },
            "mv_0693_002_a_1_1.wav":{
                "points":[11],
                "color":"#48ff00"
            },
            "mv_0693_008_C331_1_1.wav":{
                "points":[10],
                "color":"#ff006d"
            },
            "mv_0693_007_o_1_0.wav":{
                "points":[5],
                "color":"#4800ff"
            }
        },
        "Phoneme":{
            "__filterable":true,
            "h":{
                "Points":[0, 8],
                "color":"#00ff3f"
            },
            "k":{
                "Points":[1, 6],
                "color":"#003fff"
            },
            "v":{
                "points":[7],
                "color":"#ff00bf"
            },
            "C331":{
                "Points":[3, 10],
                "color":"#ff0000"
            },
            "o":{
                "Points":[5, 12],
                "color":"#7f00ff"
            },
            "a":{
                "Points":[2, 9, 11],
                "color":"#7fff00"
            },
            "i":{
                "points":[4],
                "color":"#00ffff"
            }
        },
        "Stress":{
            "__filterable":true,
            "stressed":{
                "Points":[6, 8, 10, 11, 12],
                "color":"#00ff00"
            },
            "unstressed":{
                "Points":[0, 1, 2, 3, 4, 5, 7, 9],
                "color":"#0000ff"
            }
        },
        "Voice":{
            "__filterable":true,
            "voiced":{
                "Points":[2, 3, 4, 5, 7, 9, 10, 11, 12],
                "color":"#0000ff"
            },
            "unvoiced":{
                "Points":[0, 1, 6, 8],
                "color":"#00ff00"
            }
        }
    }
}
```

Metadata csv format

n * m table with the first column containing file names.
The first row contains column headers.
Each row of data has to contain the same number of columns as the header row.

Below is an example of a working metadata .csv file:

```
file name,phoneme,voice,stress
mv_0763_019_a_1_1.wav,a,voiced,stressed
mv_0768_041_C230_1_0.wav,C230,voiced,unstressed
mv_0750_118_t_0_0.wav,t,unvoiced,unstressed
mv_0772_049_t_0_1.wav,t,unvoiced,stressed
mv_0735_060_i_1_1.wav,i,voiced,stressed
mv_0769_055_i_1_0.wav,i,voiced,unstressed
mv_0721_009_t_0_1.wav,t,unvoiced,stressed
mv_0721_027_t_0_0.wav,t,unvoiced,unstressed
mv_0726_011_C230_1_0.wav,C230,voiced,unstressed
mv_0747_015_s_0_0.wav,s,unvoiced,unstressed
mv_0752_018_i_1_1.wav,i,voiced,stressed
mv_0741_003_k_0_1.wav,k,unvoiced,stressed
mv_0703_020_u_1_0.wav,u,voiced,unstressed
mv_0756_005_m_1_1.wav,m,voiced,stressed
mv_0709_007_n_1_1.wav,n,voiced,stressed
mv_0758_024_d_1_0.wav,d,voiced,unstressed
mv_0745_036_l_1_1.wav,l,voiced,stressed
mv_0772_001_o_1_1.wav,o,voiced,stressed
mv_0756_006_i_1_1.wav,i,voiced,stressed
mv_0772_065_s_0_0.wav,s,unvoiced,unstressed
mv_0764_032_a_1_1.wav,a,voiced,stressed
mv_0764_028_e_1_1.wav,e,voiced,stressed
mv_0737_034_l_1_0.wav,l,voiced,unstressed
mv_0715_011_s_0_1.wav,s,unvoiced,stressed
mv_0732_026_i_1_0.wav,i,voiced,unstressed
mv_0763_050_s_0_0.wav,s,unvoiced,unstressed
mv_0750_021_i_1_0.wav,i,voiced,unstressed
mv_0763_006_s_0_1.wav,s,unvoiced,stressed
mv_0728_100_s_0_1.wav,s,unvoiced,stressed
mv_0705_005_n_1_1.wav,n,voiced,stressed
mv_0742_091_n_1_1.wav,n,voiced,stressed
mv_0752_025_j_1_1.wav,j,voiced,stressed
mv_0766_007_C230_1_0.wav,C230,voiced,unstressed
```

And an example of special cases:

```
heading_1,heading_2,heading_3,heading_4,heading_5
"Text containing "" and ,",1,value,,
```

The csv is equivalent to the table below:

heading_1	heading_2	heading_3	heading_4	heading_5
Text containing “ and ,	1	value		

Quotes in text are escaped with quotes.
Values containing commas have to be surrounded by quotes.
Empty fields must be explicitly be left empty. I.E. trailing commas are required for empty fields.

This format is defined to be easy to read and write using the built in csv reader/writer in python.

Fingerprint import format

Fingerprint loading is supported for csv, tsv and numpy files.

CSV

Comma delimited text files with file name in the first column and values in the following columns. No header.

```
File_1,1.27,2.55,5.67,6.77 ...
File_2,2.27,3.55,6.67,7.77 ...
```

TSV

Tab delimited text files with file name in the first column and values in the following columns. No header.

```
File_1 1.27   2.55   5.67   6.77 ...
File_2 2.27   3.55   6.67   7.77 ...
```

Numpy format

Binary format read and written by numpy.
The data itself is a list of tuples that contain file names and numpy.ndarray objects.

That is, for example:

```
[("file_1", numpy.asarray[1.27, 2.55, 5.67, 6.77]),
 ("file_2", numpy.asarray[2.27, 3.55, 6.67, 7.77])]
```

This data can be written with `numpy.save` and written with `numpy.load`.