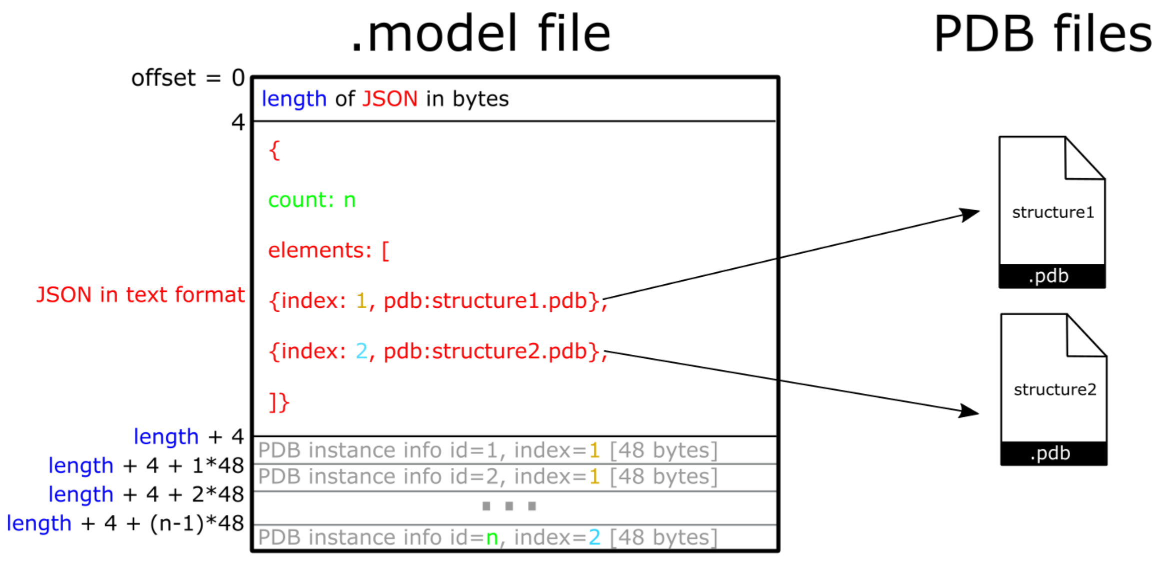
Model file format specification

One of the possible outputs of MesoCraft is .model file format. This file contains information about the current model and it's hierarchy. The description of the file format follows.

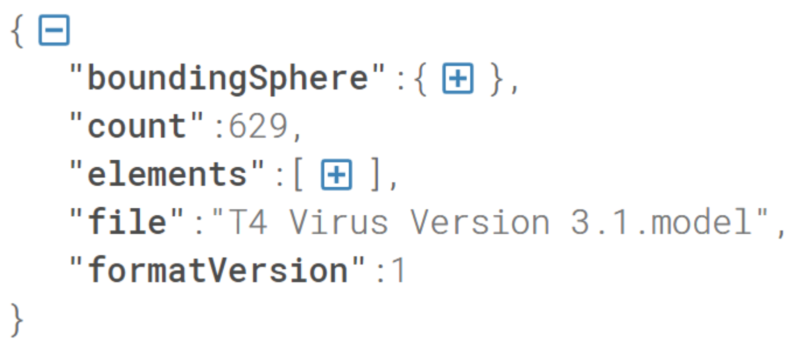
Model file format is a mixed JSON text header and binary data file. The first 4 bytes specify the length of the JSON information (metadata describing the model). Following these 4 bytes, JSON is stored, following by the binary data (12*4 bytes per instance). The instance binary data represents one instance of structure loaded from a PDB in the scene. PDB files are not part of the file and are expected to be parsed separately and centered to the origin before application of transformations from .model file. Overall file structure can be seen in the following schematic drawing.


.model file
PDB files
offset = 0
4
length of JSON in bytes
{
count: n
elements: [
{index: 1, pdb:structure1.pdb},
{index: 2, pdb:structure2.pdb},
]}
JSON in text format
length + 4
length + 4 + 1*48
length + 4 + 2*48
length + 4 + (n-1)*48
PDB instance info id=1, index=1 [48 bytes]
PDB instance info id=2, index=1 [48 bytes]
PDB instance info id=n, index=2 [48 bytes]
structure1
.pdb
structure2
.pdb

JSON specification (v. 1)

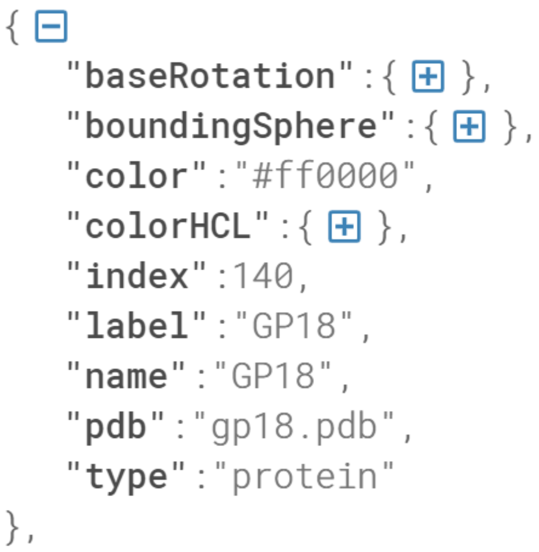
JSON header contains information about the number of elements of the model, their referencing PDB etc. The description follows:

The root node contains the amount of element (count - parameter necessary for parsing the binary data), name of the model and further metadata about the elements.

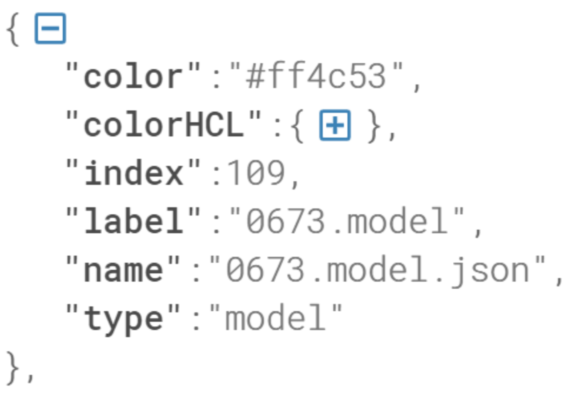


```
{ ⊟
    "boundingSphere":{ ⊞ },
    "count":629,
    "elements":[ ⊞ ],
    "file":"T4 Virus Version 3.1.model",
    "formatVersion":1
}
```

In the *elements* array, metadata regarding every type (not instance!) are included. There are two main types of elements - *protein* and *model*. In the case of *protein* type, there is an attribute *pdb* this type refers to.



```
{ ⊟
    "baseRotation":{ ⊞ },
    "boundingSphere":{ ⊞ },
    "color":"#ff0000",
    "colorHCL":{ ⊞ },
    "index":140,
    "label":"GP18",
    "name":"GP18",
    "pdb":"gp18.pdb",
    "type":"protein"
},
```



```
{ ⊟
    "color":"#ff4c53",
    "colorHCL":{ ⊞ },
    "index":109,
    "label":"0673.model",
    "name":"0673.model.json",
    "type":"model"
},
```

Binary instance info

Is formed by 48 bytes data chunk with the following layout:

[4 bytes][int] index
[4 bytes][float] position x
[4 bytes][float] position y
[4 bytes][float] position z
[4 bytes][float] rotation x
[4 bytes][float] rotation y
[4 bytes][float] rotation z
[4 bytes][float] rotation w
[4 bytes][int] id
[4 bytes][int] parent id
[4 bytes] <reserved 1>
[4 bytes] <reserved 2>

Parameters:

index - reference to the index field in the element node in the JSON metadata (can be understood as *type id*)

position - 3D vector

rotation - specified as a quaternion

id - unique identifier of the instance of the given type. Thus, proteins and models have both unique number sequence (i.e. there might exist a *protein* and a *model* with the similar id - the type needs to be taken into account while parsing the file)

parent id - id of a parent. Parent is always of type *model* (i.e. a parent cannot be a *protein*).