

Adding new metrics

- **FastQC uploader**

The parser are used to parse FastQC, Samtools and Biotoools to process them. The parser are made to fully process the whole .txt file as long it follows the same format in the file. You will need to add the metric according to the correct name from the files to the MetricIdentifier list in Main. To add a classification you will need to assign a classifier method in the Analyser class, new analysisation methods can of course be added. Finally there is also an option to add a modifier to the given data using a MetricManipulator created in the ChartBuilder class.

- **Database**

The database is MySQL and is used to store data. For example the metrics and samples. The first step to adding a new metric to the database is adding a new table to the buildDatabase.sql.

```
CREATE TABLE `ultraqc_db`.`metric_per_sequence_quality_scores` (  
  `SampleID` INT NOT NULL,  
  `Quality` INT NOT NULL,  
  `Count` DOUBLE NULL,  
  PRIMARY KEY (`SampleID`, `Quality`),  
  CONSTRAINT `QualScoreID`  
    FOREIGN KEY (`SampleID`)  
    REFERENCES `ultraqc_db`.`samples` (`ID`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

This table will store the chart data and a foreign key referring to the sample it belongs to in the sample table. In this sample table you can add a column for the classification of the new metric. To support this column you will need to tweak the QUERY.INSERTION.INSERT query by adding the new field.

You will need to add an insertion query to QUERY.INSERTION in the database.js to insert the metric data in the newly created metric table.

```
per_sequence_quality_scores: "INSERT INTO  
`ultraqc_db`.`metric_per_sequence_quality_scores` (`Quality`, `Count`, `SampleID`)  
VALUES ?"
```

A selection query can also be created by adding a new metric object to QUERY.SELECTION in database.js.

```
per_sequence_quality_scores: query({  
  table: "metric_per_sequence_quality_scores", metric:  
  "per_sequence_quality_scores",  
  x: "Quality", y: "Count"  
})
```

- **Node server**

You will need to add the correct metric name to the list of supported metrics in the app.js, so that the server will accept the new data.

- **Config**

The config files are used to store metric meta information. To add a new metric a new 'metric' needs to be added with a range. The range should show when it is bad (BAD) and when it is good (GOD). More information has been described in the config section. Note that both the client and server configs should be updated.

Config files

- **Client**

This config file can be used for several modifications to the charts. This is an example of an object in metrics:

```
"per_sequence_gc_content": {
  "display": "Normalized GC content",
  "tooltip": "Normalized GC content. A distribution graph showing the GC content of each sample, normalized around the origin."
}
```

The "display" value is the title displayed above the graph. The "tooltip" value is the tooltip which shows when hovering over the small question mark next to a graph.

This is an example of an object in charts:

```
"per_base_n_content": {
  "xAxis": { "title": { "text": "Position in read (bp)" } },
  "yAxis": {
    "title": { "text": "Percentage N-count" },
    "plotBands": [
      { "color": colours.GOOD, "from": 0, "to": 5 },
      { "color": colours.AVERAGE, "from": 5, "to": 20},
      { "color": colours.BAD, "from": 20, "to": 100}
    ],
    "min": 0
  }
}
```

The "xAxis" and "yAxis" are used for setting the text on the axes. For changing the background colours of a graph, the "plotBands" value can be used. The "min" is the minimum y-axis value.

- **Server**

This config file can be used for changing the classification values of the metrics. Next to that, also the colour of the lines in the charts can be changed.

This is an example of an object in metrics:

```
"per_sequence_gc_content": {  
  "range": ["BAD", [0.05, 0.1], "GOD"]  
}
```

Everything below 0.05 is classified as bad, between 0.05 and 0.1 as average and above 0.1 as good, and not as god. With those two values, the classification can be tuned.

This is the code for changing the colours:

```
"colours": {  
  "GOD": "rgba(40,167,69,0.4) ",  
  "AVG": "rgba(255,193,7,0.5) ",  
  "BAD": "rgba(220,53,69,0.5) ",  
  "HIS": "rgba(1,1,1,1) ",  
  "RAN": "rgba(150,150,150,.45) "  
}
```

"GOD", "AVG", "BAD" are the lines in the charts for good, average and bad samples. "HIS" is the colour of the mean line and "RAN" is the colour of the range in the charts.