

Qiskit Experiments: New Visualization Module

Conrad Haupt

IBM Quantum Research Intern, Zürich

- `qiskit_experiments.visualization`
- In current dev. version, to be in v0.5.0.
 - PR [#902](#) and [#948](#).
- Two important classes:
 - Plotters: Take in data, interact with drawers, return a figure.
 - Drawers: Create figure canvas, add graphics, return a figure. (Matplotlib)

- `qiskit_experiments.visualization`
- In current dev. version, to be in v0.5.0.
 - PR [#902](#) and [#948](#).
- Two important classes:
 - **Plotters:** Take in data, interact with drawers, return a figure.
 - **Drawers:** Create figure canvas, add graphics, return a figure. (Matplotlib)

Visualization Module



- `qiskit_experiments.visualization`
- In current dev. version, for v0.5.0.
 - PR [#902](#) and [#948](#).
- Two important classes:
 - Plotters: Take in data, interact with drawers, return a figure.
 - Drawers: Create figure canvas, add graphics, return a figure. (Matplotlib)

Curve Plotting

- For Curve Analysis.
 - Fits a model to measurements.
 - Parameter fits are results from analysis.

Curve Plotting

- For Curve Analysis.
 - Fits a model to measurements.
 - Parameter fits are results from analysis.
- Data split into
 - *series*: “data associated with a legend entry”,
 - *supplementary*: “associated with the figure, not a legend entry”.
 - Both identified by a data-key: e.g., “x” and “y”.
 - *Series* data also identified by a series-name.

Curve Plotting

- For Curve Analysis.
 - Fits a model to measurements.
 - Parameter fits are results from analysis.
- Data split into
 - *series*: “data associated with a legend entry”,
 - *supplementary*: “associated with the figure, not a legend entry”.
 - Both identified by a **data-key**: e.g., “x” and “y”.
 - *Series* data also identified by a **series-name**.

Curve Plotting

- For Curve Analysis.
 - Fits a model to measurements.
 - Parameter fits are results from analysis.
- Data split into
 - *series*: “data associated with a legend entry”,
 - *supplementary*: “associated with the figure, not a legend entry”.
 - Both identified by a data-key: e.g., “x” and “y”.
 - *Series* data also identified by a series-name.
 - **Data-keys are “unique” to a plotter class.**

Notebook Demo



Analysis Classes

```
# Inside an analysis class.

def _run_analysis(self):
    ...

    if self.options.plot:
        ...

        self.plotter.set_series_data(series_name, ...)
        ...

    ...
    if self.options.plot:
        return analysis_results, [self.plotter.figure()]
    return analysis_results, []
```

```
# Before the visualization module.

def _run_analysis(self):
    ...

    if self.options.plot:
        ...

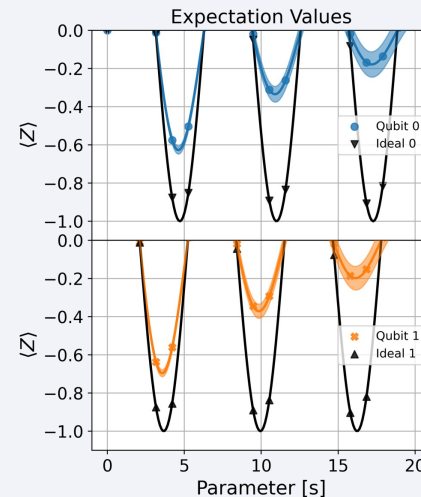
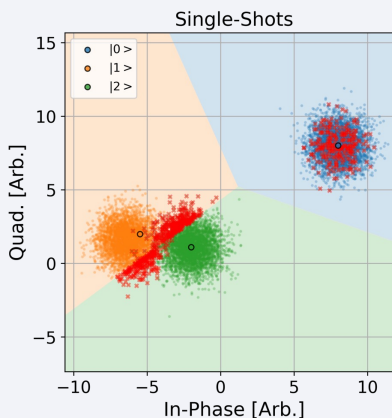
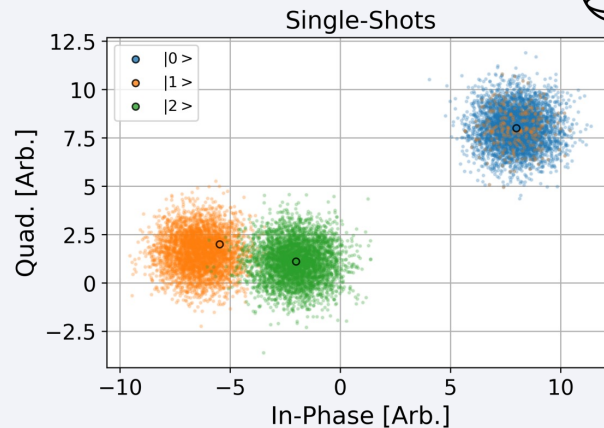
        # Not reusable and mixed with analysis code.
        self.drawer.draw_raw_data(...)

        # Some analysis-related code.
        self._some_analysis_processing(...)
        ...

    ...
    if self.options.plot:
        return analysis_results, [self.drawer.figure]
    return analysis_results, []
```

Integration with QE

- In CurveAnalysis subclasses.
 - Cross-Resonance Experiments.
 - Rabi, T1, T2Ramsey.
 - Spectroscopy.
- Plotters (CurvePlotter and IQPlotter).
 - Reusable and customizable.
 - Backend agnostic.
 - Versatile framework for QE.



Feature Goals

- Additional plotting classes.
- Expansion of options and features for existing plotter classes.
- Integration into more of Qiskit Experiments.

Contact

Qiskit Slack: @conrad

GitHub: @conradhaupt