quanp

Release 0.1

May 16, 2022

Contents

Quanp is a scalable toolkit for analyzing cross-sectional and longitudinal/time-series quantitative data. It was first inspired by scanpy and jointly built with anndata. It includes preprocessing, visualization, clustering, features selection/importance.

Read the documentation. If you'd like to contribute by opening an issue or creating a pull request, please take a look at our contributing guide. If Quanp is useful for your research, consider being a contributor.

- Discuss usage on Discourse and development on GitHub.
- Get started by browsing *tutorials*, *usage principles* or the main API.
- Follow changes of AnnData in the release notes.

CHAPTER 1

News

CHAPTER 2

Latest additions

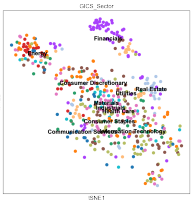
2.1 Tutorials

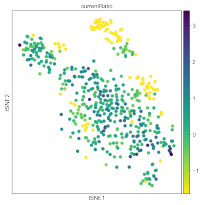
2.1.1 Clustering

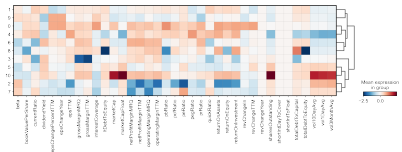
For getting started, we recommend Quanp's implementations for S&P500 member companies that contain preprocessing, clustering and the identification of features that defined a group/cluster of companies.

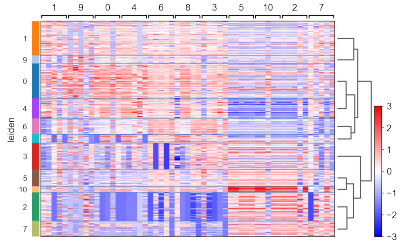
[2020 Aug] CharacterisingCompaniesBasedOnFinancialMetricsDuringCovid19
Medium Post is also available at https://medium.com/analytics-vidhya/characterising-companies-based-on-financial-metrics-during-covid19-1a6ce9cc4ada

[2020 Dec] CharacterizingOutperformedCompanies1MonthPostCOVID19VaccineSuccess
Medium Post is also available at https://quantitative-python.medium.com/characterizing-outperformed-companies-1-month-post-covid19-vaccine-success-d03185e167a9

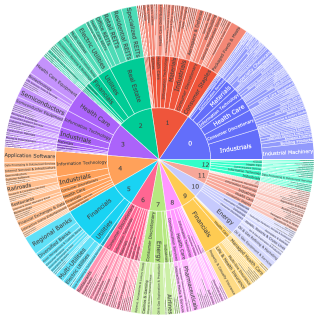








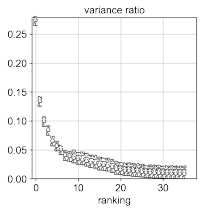
Sunburst chart

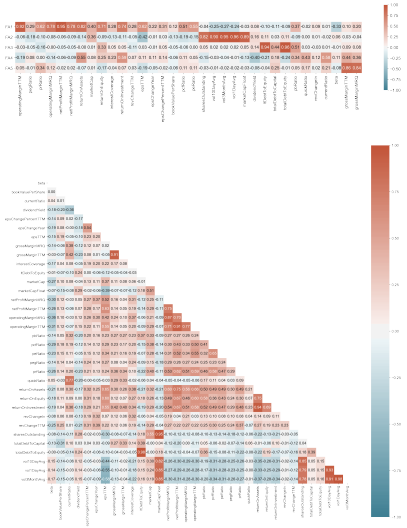


2.1.2 Factor Analysis

This tutorial analysed and visualized the underlying features that explain each principle component/factor extracted from the S&P500 member companies.

[2020 Sep] FactorAnalysisForCompaniesBasedOnFinancialMetricsDuringCovid19
Medium Post is also available at
https://medium.com/swlh/factor-analysis-characterising-companies-based-on-financial-metrics-3d5fcc4e8b6f





2.2 Usage Principles

Import Quanp as:

```
import quanp as qp
```

2.2.1 Workflow

The typical workflow consists of subsequent calls of data analysis tools in *qp.tl*, e.g.:

```
qp.tl.umap(adata, **tool_params)  # embed a neighborhood graph of the data using UMAP
```

where *adata* is an `AnnData` object. Each of these calls adds annotation to an expression matrix *X*, which stores *n_obs* observations (subjects) of *n_vars* variables (features). For each tool, there typically is an associated plotting function in *qp.pl*:

```
qp.pl.umap(adata, **plotting_params)
```

If you pass *show=False*, a `Axes` instance is returned and you have all of matplotlib's detailed configuration possibilities.

To facilitate writing memory-efficient pipelines, by default, Quanp tools operate *inplace* on *adata* and return *None* – this also allows to easily transition to out-of-memory pipelines. If you want to return a copy of the `AnnData` object and leave the passed *adata* unchanged, pass *copy=True* or *inplace=False*.

2.2.2 AnnData

Quanp is based on `anndata`, which provides the `AnnData` class.

At the most basic level, an `AnnData` object *adata* stores a data matrix *adata.X*, annotation of observations *adata.obs* and variables *adata.var* as *pd.DataFrame* and unstructured annotation *adata.uns* as *dict*. Names of observations and variables can be accessed via *adata.obs_names* and *adata.var_names*, respectively. `AnnData` objects can be sliced like dataframes, for example, *adata_subset = adata[:, list_of_feature_names]*. For more, see this blog post.

To read a data file to an AnnData object, call:

```
adata = qp.read(filename)
```

to initialize an AnnData object. Possibly add further annotation using, e.g., *pd.read_csv*:

```
import pandas as pd
anno = pd.read_csv(filename_sample_annotation)
adata.obs['subject_groups'] = anno['subject_groups']  # categorical annotation of␣
↪type pandas.Categorical
adata.obs['time'] = anno['time']                # numerical annotation of type float
# alternatively, you could also set the whole dataframe
# adata.obs = anno
```

To write, use:

```
adata.write(filename)
adata.write_csvs(filename)
adata.write_loom(filename)
```

2.3 Installation

2.3.1 Anaconda

If you do not have a working installation of Python 3.6, consider installing Anaconda with Python=3.6 and create a vitualenv using conda. Then run:

```
conda install seaborn scikit-learn statsmodels numba pytables
conda install -c conda-forge python-igraph leidenalg
```

The extra *python-igraph* and *leidenalg* installs two packages that are needed for popular parts of quanp but aren't requirements: python-igraph [Csardi06] and leiden [Traag18].

Pull Quanp from PyPI (consider using `pip3` to access Python 3):

```
pip install quanp
```

2.3.2 Development Version

To work with the latest version on GitHub: clone the repository and *cd* into its root directory. To install using symbolic links (stay up to date with your cloned version after you update with *git pull*) call:

```
pip install -e .
```

2.3.3 Troubleshooting

If you get a *Permission denied* error, never use *sudo pip*. Instead, use virtual environments or:

```
pip install --user quanp
```

On MacOS, if **not** using *conda*, you might need to install the C core of igraph via homebrew first

- *brew install igraph*

- If python-igraph still fails to install, see the question on compiling igraph. Alternatively consider installing gcc via *brew install gcc –without-multilib* and exporting the required variables:

```
export CC="/usr/local/Cellar/gcc/X.x.x/bin/gcc-X"
export CXX="/usr/local/Cellar/gcc/X.x.x/bin/gcc-X"
```

  where *X* and *x* refers to the version of *gcc*; in my case, the path reads */usr/local/Cellar/gcc/6.3.0_1/bin/gcc-6*.

On Windows, there also often problems installing compiled packages such as *igraph*, but you can find precompiled packages on Christoph Gohlke's unofficial binaries. Download those and install them using *pip install ./path/to/file.whl*

2.3.4 Installing Anaconda

After downloading Anaconda, in a unix shell (Linux, Mac), run

```
cd DOWNLOAD_DIR
chmod +x Anaconda3-latest-VERSION.sh
./Anaconda3-latest-VERSION.sh
```

and accept all suggestions. Either reopen a new terminal or *source ~/.bashrc* on Linux/ *source ~/.bash_profile* on Mac. The whole process takes just a couple of minutes.

2.4 References

Bibliography

[Csardi06] Csardi *et al.* (2006), *The igraph software package for complex network research*, InterJournal Complex Systems.

[Traag18] Traag *et al.* (2018), *From Louvain to Leiden: guaranteeing well-connected communities* arXiv.