Claire McQuin

425-244-8478 mcquincl@gmail.com Somerville, MA

4+ of experience maintaining and developing modular, extensible and flexible open source software enabling users to create solutions across a variety of domains, including configuration management and biological research.

Skills

- Maintaining, developing and distributing Ruby and Python applications for Linux, OS X, and Windows platforms;
- Developing full-stack Erlang/OTP applications with Javascript and Angular UI, backed by PostgreSQL databases;
- Defining and expanding APIs for system configuration and visibility;
- Developing applications for deep learning in Python with Keras;
- Additional experience with R, Java, Scala and C/C++.

Experience

Software Engineer *Broad Institute of MIT and Harvard, Cambridge, MA* **July 2016 to Present**

Collaborating with biologists and statisticians around the world to enable discoveries in biology and medicine by:

- maintaining and contributing to open-source applications for extracting features from cellular microscopy images ("CellProfiler") and downstream analysis of high-throughput microscopy experiments ("cytominer");
- implementing support for processing and analyzing three-dimensional single-channel microscopy images;
- creating a tutorial on three-dimensional image analysis which has been adopted by a leading image processing library ("scikit-image") and presented by their team at several major scientific computing conferences;
- developing a generalized application for single-cell phenotype classification using deep residual networks.

Improving the quality of software released by the lab and reducing maintenance overhead by using automation to:

- continuously integrate new developments on Linux and Windows platforms;
- build and deploy packages to the Python Package Index, and build and publish documentation;
- perform daily end-to-end testing of common workflows to monitor compatibility with nightly software releases.

Software Development Engineer *Chef Software (formerly Opscode), Seattle, WA* **April 2014 to June 2016**

Designed and developed new features for Chef's automation software. Empowered organizations to rapidly and safely develop their software and infrastructure by implementing features which:

- detect failures in dependent applications before promoting software to production via automatic dependency testing;
- decrease development cycle time by alerting contributors when changes are ready to be reviewed or promoted, decrease time to detection of defects, and provide insight into the health of their release pipeline through notifications sent to often-used notifiable services;
- run compliance profiles on deployments, upgrades, and changes to ensure infrastructure passes audit requirements before official audits are conducted.

Reduced defects in releases of Chef's primary open source software project through automated end-to-end testing of core capabilities on every contribution. Maintained Chef's open source software projects by:

- triaging opened issues and linked duplicates, assigned appropriate labels for easy filtering, answered questions, and added new issues to the team's backlog;
- reviewing contributions, provided feedback and assistance as needed, and merged once completed;
- packaging and releasing updates and communicated releases to the community.

Continued to support and develop the "Ohai" project (an open source, plugin-based system profiling application rewritten during my internship) as a community maintainer:

- provided users more control over the application by implementing configuration file support;
- augmented the existing features of the application's domain specific language to provide safer data manipulation;
- attended weekly community IRC meetings and participated in the community's RFC process for new features;
- lead open space discussions at community events to present new features, answer questions, receive feedback and discuss improvements.

Software Development Intern *Opscode, Seattle, WA* **Summer 2013**

Implemented updates to an open source, plugin-based system profiling application ("Ohai"). Provided plugin authors finer control over which platforms their plugins collect data on and improved plugin organization:

- added new methods to the application's Ruby domain specific language (DSL);
- refactored the application to support the DSL updates and updated core plugins to use the new DSL;
- maintained backward compatibility with the previous versions of the DSL to encourage users to upgrade to the latest version of the software without significant investments of time and effort.

Education

Bachelor of Science, Computer Science and **Mathematics** **Winter 2014**

University of Washington, Seattle, WA

https://www.linkedin.com/in/claire-mcquin-33690965/