NumFOCUS John Hunter Fellowship Proposal cover page

Applicant
Olga Botvinnik
4067 Miramar St
Apt A
La Jolla, CA 92037
obotvinn@ucsd.edu
541-953-2482

Scientific Mentor
Gene Yeo, PhD MBA
2880 Torrey Pines Scenic Dr
La Jolla, CA 92037
geneyeo@ucsd.edu
(858) 453-9321

Software Mentor
C. TItus Brown, PhD
2215 Biomedical and Physical Sciences
567 Wilson Road
Michigan State U.
East Lansing, MI 48824
ctb@msu.edu

Proposed start date: Monday, June 30th, 2014 (flexible)

In an ideal world, my day would consist of going over yesterday's results, trying out some new analyses, writing an IPython Notebook blog post about today's work, and communicating with other members of my field through blog posts, twitter, and a weekly Google+ Hangout live video-chat lab meeting. However, currently, similar research is performed behind closed doors, duplicating data and analyses that could have been openly shared. For example, thousands of groups around the world perform RNA-Seq to quantify gene expression, and each one of them performs the same optimization on both the experimental and computational side, and these common tasks would be much easier if even a few groups posted a few short blog posts on their troubleshooting. This way, the method is front and center, rather than buried in the back of the supplementary section of paper. In particular, my field of molecular biology is far behind other data-intensive biological fields such as genomics, where the prisoner's dilemma of data sharing was long solved by creating sharing consortiums by the National Institutes of Health (NIH) for genome-wide association studies. **Through the NumFOCUS John Hunter Fellowship, I will pioneer data and code sharing in the single-cell and RNA biology worlds.**

My field of RNA biology has recently exploded through the ability to measure genes in single cells. Almost all previous biological experiments were performed by grinding up hundreds of thousands of cells and getting an average measurement, say of gene expression. Just as surveying the mean eye color of a conference's attendees doesn't tell you the actual diversity, only the average, **I study how differences individual cells create the diversity of a whole.** Specifically, I study how different versions of a gene are created by cutting and pasting different substrings of the same DNA together, through a process called alternative splicing (Kornblihtt et al. 2013; Cáceres and Kornblihtt 2002). What we don't know is how these different versions are distributed between individual cells. This is especially important in studying the brain and spinal cord because there are so many different kinds of cells (Yeo et al. 2004), and each one of them has its own repertoire of favorite versions of genes to use, much like every coder has their own personal setup, with their own unique combination of text editor, debugger, coding style, naming schemes, and so on. We study how individual skin-derived pluripotent, neural progenitor, and motor neuron cells express different versions of genes, and how the distributions of versions (there could be hundreds or even thousands) change between population (Tietjen et al. 2003; Yu, Marchetto, and Gage 2013; Venables et al. 2013). This research has important future implications in understanding rare cell populations, and **applications such as early cancer detection via trace signals in blood, pre-natal monitoring via fetal cells in maternal blood, and teasing apart neurological conditions such as autism, Alzheimer's disease, and neurodegenerative diseases such as Amyotrophic lateral sclerosis (ALS, also known as Lou Gehrig's disease)** (Marinov et al.; Wen and Tang 2014; Junker and van Oudenaarden 2014; Nawy 2014; Macaulay and Voet 2014; Chattopadhyay et al. 2014).

The results of my research will be packaged to **create open-source software for the single-cell community.** The analyses will focus on creating Python tools for use in the Interactive Python notebook (Pérez and Granger 2007). Currently, the only widespread methods for basic analysis of single-cell data are proprietary and closed-source, stunting the growth of the field. All methods and software described will be distributed as open-source under the MIT license, which allows for anyone to remix and reuse the software as they please, but does not provide any warranty. By providing the community with user-friendly basic analyses and our advanced alternative splicing analyses, **we will unify the community to develop analyses together, and advance the field by preventing the need for each researcher to "reinvent the wheel," and instead focus on novel analyses.** Additionally, we will share our data before the paper is published, and with our papers, release an IPython Notebook detailing the analyses and creation of each figure.

Already, I have contributed to several open source projects. Passionate about data visualization, I created `prettyplotlib` ("Olgabot/prettyplotlib" 2014), a wrapper around the Python plotting library `matplotlib` (Hunter 2007), and I gave a lightning talk on this software at PyData NYC 2013. I have also contributed to the `seaborn` library, a statistical visualization library built on top of `matplotlib`, `statsmodels`, and other scientific packages. Additionally, I create and release small pieces of code which accomplish a small task, such as the program `sj2psi` , which uses an output file describing splice junctions ("sj") from an alignment program to calculate alternative splicing scores ("percent spliced in" or "psi" scores). While it is a tiny piece of code (129 lines), it is still a useful tool for those interested in these calculations. One person from another biological lab across the country in Boston is "watching" the repo, and his research will likely benefit whenever I make updates.

Open-source projects that would directly benefit from this work:

- `seaborn ("Mwaskom/seaborn" 2014)`: An extension of matplotlib, developed initially for visualizing linear relationships between variables by a computational neuroscience graduate student, but I have incorporated the nonlinear clustering analyses that I perform into the package, and have drafted code for quickly plotting results of dimensionality reduction algorithms from scikit-learn. My research makes extensive use of dimensionality reduction algorithms, and I would like to help others interested in trying out the algorithm so they may quickly iterate on their analyses, rather than spend time figuring out plotting parameters.
- `sj2psi ("olgabot/sj2psi" 2014)`: Tool I created for calculating percent spliced-in ("psi") scores. It calculates these agnostically, without knowledge of the gene structure, unlike other programs. Currently this program is very bare-bones and through my research, I will continue to add more advanced functionality.

Open source projects that would be used by this work, and possibly be contributed to:

- **IPython notebook** (IPyNB) (Pérez and Granger 2007): I do all my analysis and figure generation exclusively in IPyNBs (ssh tunneled into our supercomputer), because it is so simple to iterate on figure generation and share analyses with collaborators. Recently, my colleague used the interactive widgets of IPython to create an interactive principal component analysis plot of gene expression. This has been very successful in the lab because the wet-bench biologists can now explore the data with specific lists of genes they may be interested, and perform their own exploratory analyses. **During this fellowship tenure, I will integrate my analyses of alternative splicing with IPyNB interactive widgets to produce an open-source tool for exploratory single-cell analysis**, hopefully trumping the current proprietary R implementation. We are currently beta-testing our implementations in-house with non-programmer biologists and will make our code publicly available once we've fleshed out the bugs. We will debut these analyses and tools our local single-cell user's group for further feedback, where we will make the code publicly available.
- **`matplotlib`** `(Hunter 2007):` I use this Python data visualization library every day, and it is an invaluable member of my analysis toolkit. The ability to quickly assess whether an experiment was successful or not, visualize differences in distributions, and compare samples across multiple experiments makes my work and the work in our lab able to advance much more quickly than labs using other languages.

- **pandas** (McKinney 2012)**:** Data transformation and munging are no longer a pain using pandas - instead they are a dream. The program I created, sj2psi, was possible exactly because of the ease of the groupby, apply, and transform functions in pandas. The same calculation could have been performed via large amounts of text parsing, but it became almost trivial with the use of pandas. I use pandas daily to read, filter, and format data for analysis, and will continue to use it extensively throughout the fellowship tenure.
- **Bokeh:** With the large amounts of data of single cells (150,000 gene measurements per cell, ~300 cells) it becomes important to iterate quickly in analyses. Simple interactivity such as hover tooltips help to identify outliers without the pain of replotting a figure just for the gene or sample name. I've only used Bokeh for a few plots but look forward to incorporating it into more of our interactive work.
- **gffutils** ("Daler/gffutils" 2014)**:** General Feature Format (GFF) files are a commonly used file type encoding contain genomic coordinates, annotation, and sub-feature information for each gene. But extracting these features systematically from files using text parsing becomes very convoluted for even slightly complex queries, such as getting all of the third specific subfeatures of genes from a subset of regions. The problems solved by gffutils are widespread throughout bioinformatics: data is available, but the formatting is confusing. I will use gffutils to identify differences in types of genes and gene properties associated with different celltypes.
- **scikit-learn:** I use scikit-learn (Pedregosa et al. 2011) nearly every day to reduce and cluster my data. It is an incredible valuable tool for our work. I would not have been able to perform a massive 4x6x3x5x4 parameter grid search using multiple different clustering algorithms and dimensionality reduction techniques. The standard fit and transform functions made it possible to easily test out many different algorithms. Had this package not existed, I would have had to hunt down implementations of individual algorithms or implemented them myself, losing substantial research time to development. I will continue to use scikit-learn for data exploration throughout this fellowship.

I am so thankful for the Python data analysis and scientific communities for creating such powerful tools for scientific computing.

My software engineering mentor, Dr. C. Titus Brown is an excellent fit for this project. Dr. Brown has run several successful bioinformatics software projects and understands the academic open source world. By working with him, I hope to gain Python coding practices, advice for managing open-source bioinformatics software, and mundane things such as dealing with users who are new to python but want to try this method on their data, installation issues. I've met with Dr. Brown several times, in both academic and technical settings, and have found his philosophies on how to approach computational problems and how to do good, open science in academia. His lab also does code reviews and other good software engineering practices that our lab currently does not, and it has been difficult to change the culture. Through working with Dr. Brown, I hope to learn coding practices specific to academic settings.

Dr. Gene Yeo will be an excellent scientific mentor. Dr. Yeo is my current advisor and has deep domain knowledge of the single-cell, RNA and computational molecular biology fields. Moreover, our laboratory is a hybrid lab, where I work alongside wet-bench biologists, rather than with external collaborators. This means that when I use a novel computational approach, the results will be quickly interpreted into biological context by my labmates, and the **scientific knowledge gained from my results**

will be realized much more quickly than if I worked in a computational-only lab. This hybrid model is a huge advantage as we are able to iterate on scientific problems very quickly.

I have recently become more involved in the Python communities through attending PyData NYC 2013 and PyCon 2014. At PyData, I was one of two biologists, and at PyCon, I was one of a dozen. I also attended the Strata data science conference, where I did not meet any academics. These conferences taught me new tools and methods for data analyses, workflows, and project management in Python. I greatly enjoy attending each of these conferences and meeting industry leaders, and I'm also grateful for the interactions with Pythonistas and Data Scientists in industry because they remind me that I am so lucky to work on biological problems. By talking to them, **I** am reminded of how much **I enjoy working on understanding how the molecules within us work to create functional human beings**.

As a woman in bioinformatics, my biologist friends (biology grad school is female dominated right now) see me doing bioinformatics and have started to venture into computation. I do think that being me female makes it easier for other women to ask me about computational questions. I notice that they seem to ask me first, rather than another male computational scientist. I helped in convincing several women who were considering computational careers to go all the way. For example, Paola studied Biochemistry at UCLA for undergrad, then I met her at University of California, Santa Cruz where we were studying for a MS in Bioinformatics and Biomolecular Engineering. Paola had never programmed before this program, and now she is Software Developer (newly-promoted from Quality Assurance Engineer) at SurveyMonkey. I am so proud of her transformation and inspired by her. She showed great initiative, and she and I worked off each other as constant source of positive energy and support. **By being a successful woman in computational sciences, I'm showing other women that they can do it too.**

At UCSD, undergraduate researchers are not necessarily paid. It can be very difficult to obtain funding. This filters out people who want to do science, but cannot afford to not have a job, thus reinforcing the privileged cycle of academia. **With this fellowship, I will use some of this money to fund an undergraduate student to work with me.** By paying an undergrad, we are not only providing an option to low-income students to pursue research, we are also able to screen for highly motivated and capable undergraduates. Additionally, the money will confer to the student additional responsibility to work consistently, rather than an unpaid volunteer who may not be as invested.

By obtaining this fellowship, I will be able to use this fellowship as external motivation with conservative academic collaborators as reason for open-sourcing code and publishing methods early as quick blog posts. I have completed all classes and passed my qualifying exam so I can work on the project full-time. **By funding me, NumFOCUS will support data and methods sharing in the community creating fundamental biological research which can lead to lifesaving treatments, novel methods in Python data visualization, women in science, and undergraduate research opportunities.**

Chattopadhyay, Pratip K, Todd M Gierahn, Mario Roederer, and J Christopher Love. 2014. "Single-Cell Technologies for Monitoring Immune Systems." *Nature Immunology* 15 (2). Nature Publishing Group: 128–35.

Cáceres, Javier F, and Alberto R Kornblihtt. 2002. "Alternative Splicing: Multiple Control Mechanisms and Involvement in Human Disease." *Trends in Genetics: TIG* 18 (4). Elsevier Ltd: 186–93.

"Daler/gffutils." 2014. Accessed May 12. https://github.com/daler/gffutils.

Hunter, JD. 2007. "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*. doi.ieeecomputersociety.org. http://doi.ieeecomputersociety.org/10.1109/mcse.2007.55.

Junker, Jan Philipp, and Alexander van Oudenaarden. 2014. "Every Cell Is Special: Genome-Wide Studies Add a New Dimension to Single-Cell Biology." *Cell* 157 (1). Elsevier Inc. 8–11.

Kornblihtt, Alberto R, Ignacio E Schor, Mariano Alló, Gwendal Dujardin, Ezequiel Petrillo, and Manuel J Muñoz. 2013. "Alternative Splicing: A Pivotal Step between Eukaryotic Transcription and Translation." *Nature Reviews. Molecular Cell Biology* 14 (3). Nature Publishing Group: 153–65.

Macaulay, Iain C, and Thierry Voet. 2014. "Single Cell Genomics: Advances and Future Perspectives." Edited by Nancy Maizels. *PLoS Genetics* 10 (1): e1004126.

Marinov, Georgi K, Brian A Williams, Ken Mc Cue, Gary P Schroth, Jason Gertz, Richard M Myers, and Barbara J Wold. "From Single-Cell to Cell-Pool Transcriptomes: Stochasticity in Gene Expression and RNA Splicing Supplementary Materials."

McKinney, W. 2012. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

"Mwaskom/seaborn." 2014. Accessed May 15. https://github.com/mwaskom/seaborn.

Nawy, Tal. 2014. "Gene Expresion: Single Cells Make the Tissue." *Nature Methods* 11 (4). Nature Publishing Group: 371–371.

"Olgabot/prettyplotlib." 2014. Accessed May 15. https://github.com/olgabot/prettyplotlib.

"olgabot/sj2psi." 2014. Accessed May 15. https://github.com/olgabot/sj2psi.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research: JMLR* 12. JMLR.org. http://portal.acm.org/citation.cfm?id=1953048.2078195&coll=DL&dl=ACM&CFID=422733224&CFTOKEN=93183407.

Pérez, Fernando, and Brian E Granger. 2007. "IPython: A System for Interactive Scientific Computing." *Computing in Science & Engineering* 9 (3). AIP Publishing: 21–29.

Tietjen, Ian, Jason M Rihel, Yanxiang Cao, Georgy Koentges, Lisa Zakhary, and Catherine Dulac. 2003. "Single-Cell Transcriptional Neurotechnique Analysis of Neuronal Progenitors." *Neuron* 38: 161–75.

Venables, Julian P, Laure Lapasset, Gilles Gadea, Philippe Fort, Roscoe Klinck, Manuel Irimia, Emmanuel Vignal, et al. 2013. "MBNL1 and RBFOX2 Cooperate to Establish a Splicing Programme Involved in Pluripotent Stem Cell Differentiation." *Nature Communications* 4: 2480.

Wen, Lu, and Fuchou Tang. 2014. "Reconstructing Complex Tissues from Single-Cell Analyses." *Cell* 157 (4): 771–73.

Yeo, Gene, Dirk Holste, Gabriel Kreiman, and Christopher B Burge. 2004. "Variation in Alternative Splicing across Human Tissues." *Genome Biology* 5 (10). BioMed Central Ltd: R74–R74.

Yu, Diana X, Maria C Marchetto, and Fred H Gage. 2013. "Therapeutic Translation of iPSCs for Treating Neurological Disease." *Stem Cells* 12 (6). Elsevier Inc. 678–88.