The ISME Journal (2012), 1–4

www.nature.com/ismej

COMMENTARY

Collaborative cloud-enabled tools allow rapid, reproducible biological insights

Benjamin Ragan-Kelley[1], William Anton Walters[1], Daniel McDonald[1], Justin Riley, Brian E Granger, Antonio Gonzalez, Rob Knight, Fernando Perez and J Gregory Caporaso

The ISME Journal advance online publication, 25 October 2012; doi:10.1038/ismej.2012.123

Microbial ecologists today face critical computational barriers. The rapid increase in the quantity of data acquired by modern sequencing instruments makes analysis by hand infeasible, and even software developed just a few years ago cannot scale to modern data sets. As a result, making advanced, scalable algorithms and large-scale computational resources available to end-users is necessary to advancing our understanding of microbial ecology.

One challenge many face when developing software for the first time is the gap between writing a script that can run on a single processor and writing a script that will scale to a larger cluster. A second is that knowledge required for a project is often distributed among many individuals, including software developers, subject matter experts and experts in the use of specific computer systems. Although computation can be a language that bridges many disciplines, additional 'glue' is often needed to make the requirements mutually comprehensible to diverse members of a project team.

One approach to this 'glue' is represented by IPython (Pérez and Granger, 2007), which provides tools for interactive and parallel computing that support online collaboration. The IPython notebook allows users to combine code, text (including mathematical expressions), figures, and so on, into a single document. These documents are accessed through a web browser and can be simultaneously edited by multiple collaborators. The resulting environment is analogous to Google Docs, but aimed at scientific computation. Beyond document writing, these notebooks can execute arbitrary code in the Python programming language, providing a framework where documentation, software and results are combined in one place, and code can be edited, annotated and re-run dynamically to immediately show how the results change. IPython also provides tools to run computations in parallel, with a high-level interface that eases the transition from a classic serial script to a parallel environment.

The power of the IPython approach is especially apparent when it is coupled to cloud computing, which is rapidly increasing in popularity in bioinformatics (Stein, 2010). Services such as Amazon's Elastic Compute Cloud (EC2) provide on-demand access to large-scale computational resources, allowing anyone to trade (small amounts of) money for (large amounts of) compute time. Although Amazon provides a web-based interface for management, in this project, we used the StarCluster tool (http://mit.edu/star/cluster) to automate and simplify the process of building, configuring and managing clusters of virtual machines on Amazon's EC2. Using StarCluster, we can configure a virtual cluster that includes domain-specific libraries (bioinformatics tools in our case), as well as cluster management tools and shared file system configuration, for 'one-click parallel computing'. Once the StarCluster configuration has been defined, we can start a virtual cluster in the cloud with a single command. StarCluster will ensure that the cluster nodes start up together, and correctly configured, drastically reducing the time and complexity of setting up a parallel cloud cluster.

These principles were exemplified at the recent NIH 'Cloud Computing for the Microbiome' workshop, in Boulder, CO, USA, which brought together participants with broad expertise including developers of IPython, Quantitative Insights Into Microbial Ecology (QIIME; Caporaso *et al.*, 2010) and PrimerProspector (Walters *et al.*, 2011), contributors to the Greengenes (DeSantis *et al.*, 2006) resource, and the author of StarCluster participating remotely from MIT. The IPython and StarCluster authors have backgrounds in physics, whereas the QIIME, PrimeProspector and Greengenes developers come from microbial ecology and bioinformatics; neither group had used the other's tools before this meeting. Initially, a demonstration of IPython had been planned for the workshop based on distributed matrix calculations, however, given the audience, demonstrating how IPython could help tackle a real biological problem using the cloud seemed far more desirable.

After considering several potential problems, we settled on one question of compelling interest and generalizability: as current sequencing technologies generally limit us to sequencing only certain regions of the 16S ribosomal RNA, what region is optimal for recapitulating the 16S phylogeny that would be

obtained from sequencing the full gene? Although previous studies have examined the role of the region and read length for taxonomic assignment (Wang *et al.*, 2007) and community clustering (Liu *et al.*, 2007), the region that best recaptures the phylogenetic tree reconstructed from full-length sequences has not been recently examined using the full Greengenes alignment. Intuition suggests that a longer sequence would automatically yield a better tree because more characters would be available, but this intuition had been proven wrong in other areas of community analysis. What would happen when short fragments were isolated from the alignment and used in the popular phylogeny package FastTree (Price *et al.*, 2010)?

Several key technologies were leveraged to answer this question: (a) the IPython notebook provided a rapid, collaborative environment for execution of code; (b) StarCluster provided an easy way to set up pre-configured clusters with dozens of central processing units on EC2; (c) the Python Comparative Genomics Toolkit (PyCogent) toolkit (Knight *et al.*, 2007) provided a large number of well-tested biological utility functions in Python; (d) Greengenes provided the source alignments and trees; (e) PrimerProspector provided easy ways to locate primer positions in the Greengenes alignment; and (f) QIIME provided visualization routines that could be deployed in a web browser. Ultimately, we succeeded in producing a working demonstration in a total development time of roughly 7 h. The IPython Notebook used for this analysis is 'NIH-CloudDemo (Complete)' (see 'Data availability' section).

Having achieved our educational goal of producing a practical demonstration of cloud computing, we now ask whether our example computation produced results of scientific value. We sliced the alignment of the full-length 16S ribosomal RNA to simulate sequencing of amplicons at different read lengths using a collection of popular primers (Figure 1a). A phylogenetic tree was subsequently constructed from each resulting alignment, and distances were computed between all trees and the tree calculated from the full-length alignment as Pearson correlations in tip-to-tip distances across trees (used in Figure 1; distances computed as $1 - r$) and Robinson–Foulds distances. Principal coordinates analysis was applied to the Pearson distance matrix to visualize the results, and the Mantel test was applied to confirm that similar results were achieved using the two distance measures. Coloring the results by the length of the read (Figure 1b) we see some association with the length of read for the v2 region, but essentially no association for other regions. Coloring the results instead by the location of the start point within the 16S ribosomal RNA sequence (Figure 1c), we see that the location within the sequence matters immensely. Thus, we can conclude that choosing the region of the 16S ribosomal RNA wisely is more important for reconstructing a useful phylogeny, such as that required for phylogenetically informed community distance metrics such as UniFrac (Lozupone and Knight, 2005), whereas obtaining longer reads should be treated as a secondary concern. We further illustrate this in Supplementary Figure S1, where we performed principal coordinates analysis only on distances between trees generated from the V3 to V4 regions and the full-length sequences (see the 'V3 and V4 Regions Only' notebook). This analysis shows that when working with regions of the 16S that best recapitulate phylogeny, longer reads yield trees that are more similar to the full-length trees

a

Forward primer	v regions(s)	Start	End	Read lengths (in addition to full amplicon)
27F	v2	136	1868	
	v2-v3	136	2232	150, 250, 400
	v2-v4	136	4051	
	v2-v6	136	4932	
	v2-v8	136	6426	
	v2-v9	136	6791	
349F	v3	1916	2232	
	v3-v4	1916	4051	150, 250, 400
	v3-v6	1916	4932	
	v3-v8	1916	6426	
	v3-v9	1916	6791	
515F	v4	2263	4051	
	v4-v6	2263	4932	150, 250, 400
	v4-v8	2263	6426	
	v4-v9	2263	6791	
967F	v6	4653	4932	
	v6-v8	4653	6426	150, 250, 400
	v6-v9	4653	6791	
1391F	v9	6450	6791	
Full-length	All	0	7682	

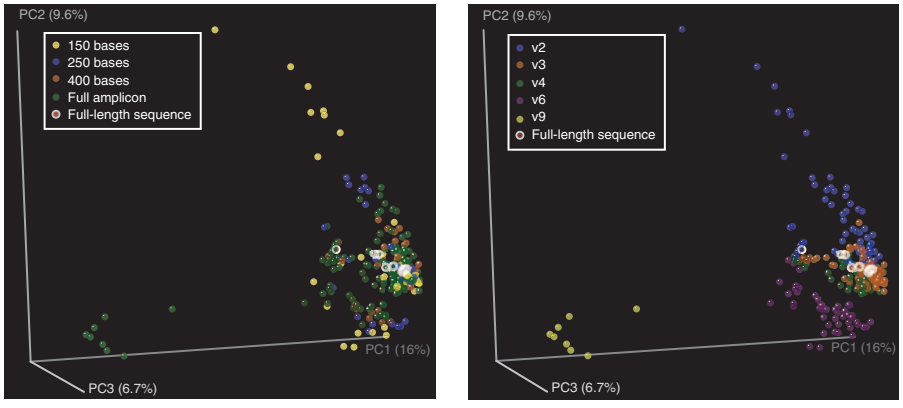

Figure 1 (**a**) Regions of the 16S ribosomal RNA included in this analysis. Start and end positions refer to positions in the Greengenes alignment and *v regions* indicate the variable regions included in each simulated amplicon. Sliced amplicons that would overlap entirely with other sliced amplicons are not included. Only full-length reads were used in analysis of the V9 region as the full-length amplicon is shorter than 150 bases. (**b**) Principal coordinates analysis of Pearson correlation coefficients between tip-to-tip distances in pairs of phylogenetic trees constructed from differentially sliced alignments. Points are colored by amplicon length. Points representing trees generated from full-length sequences are circled in white to indicate their position when obscured by other points. (**c**) Principal coordinates analysis of Pearson correlation coefficients between tip-to-tip distances in pairs of phylogenetic trees constructed from differentially sliced alignments. Points are colored by the first variable region encountered in the differentially sliced alignments. Points representing trees generated from full-length sequences are circled in white to indicate their position when obscured by other points.

than shorter reads (Supplementary Figure S1a). Finally, in the 'Pearson v Robinson–Foulds Distances' notebook, we compare Pearson distances to Robinson–Foulds distances and show that these distance are significantly correlated (Mantel test: $r = 0.77$, $P < 0.001$) as are the principal coordinates analysis plots generated from each distance matrix (Procrustes test: M^2:0.67, $P < 0.001$). The analyses presented here are easily generalizable: the user can substitute in any input alignment (for example, fungal internal transcribed spacer). The 'Variable Region Position Boundaries' notebook describes this process (see 'Data availability' section).

We emphasize how this effort generated two outcomes that facilitate validation and replication of our results: both the IPython notebooks we developed and the Amazon Machine Image (AMI) that contains all the necessary biological libraries and IPython/StarCluster support are publicly available (see 'Data availability' section). This allows anyone with an Amazon account to repeat our analysis or modify it to address related questions. The cost of the analysis depends on the size of the data set. Using an input alignment with 636 sequences (that is, Greengenes clustered into 82% operational taxonomic units), the cost is $7.40 and the runtime is 5 min on four m2.4 × large instances (the majority of the cost results from having to pay for a full hour of instance time). The complete analysis used a variety of input alignments (Greengenes operational taxonomic units ranging from 76% to 99%, roughly in steps of 3%, with between 121 and 84 413 sequences, respectively) and cost approximately $180 with a runtime of 25 h on four eight-core/68 GB-RAM instances (that is, the Amazon Web Services m2.4 × large instance type). Had this analysis not been run it parallel, the results would have required over a month to compute. The 'Timing' notebook contains additional details (see 'Data availability' section).

In conclusion, we have shown how a team of researchers with radically different backgrounds can leverage cloud resources and open source tools to achieve a new and scientifically interesting result relevant to an important question in microbial ecology, in record time and all the while producing easily reproducible outcomes. Central to this effort was the use of cloud resources not only to command and deploy a large amount of compute power, but also as an integral part of the development process itself: the team edited the IPython notebooks for this study directly on the cloud servers. This enabled multiple authors to rapidly evolve the initial draft, with each person focusing on a different aspect of the overall computation. As the shared environment provided by the IPython notebook includes code and execution results, the team could rapidly reach a mutual understanding using the shared language of computation and, through rapid, iterative development and visualization processes, achieve the desired result in the same environment meant to perform the final production runs. Cloud-enabled tools thus allow broadly applicable solutions to interesting scientific problems to be rapidly formulated, communicated and reproduced. Microbial ecologists are poised to take advantage of these advances in scientific computing by using tools like the QIIME/StarCluster/IPython pipeline described here, other existing tools such as Galaxy/CloudMan or CloudBioLinux (Afgan *et al.*, 2012), or many new tools that will come online in the coming months and years.

Data availability

The IPython notebooks and all data files referenced here are available at http://qiime.org/home_static/nih-cloud-apr2012/. The Amazon Machine Identifier used for these analyses is ami-9f69c1f6. Tutorials for using QIIME are available at http://www.qiime.org; tutorials for using the IPython notebook are available at http://ipython.org/ipython-doc/rel-0.13/index.html and http://ipython.org/videos.html; tutorials for using StarCluster are available at http://web.mit.edu/star/cluster/.

Acknowledgements

This work arose out of the NIH-supported 'Cloud Computing for the Microbiome' workshop, Boulder, CO, USA, 2–4 April 2012, and was supported in part by the NIH, the Crohn's and Colitis Foundation of America, Amazon Web Services, and the Howard Hughes Medical Institute.

B Ragan-Kelley is at Graduate Group in Applied Science and Technology, University of California at Berkeley, Berkeley, CA, USA
W A Walters is at Department of Molecular, Cellular and Developmental Biology, University of Colorado at Boulder, Boulder, CO, USA
D McDonald is at Biofrontiers Institute, University of Colorado at Boulder, Boulder, CO, USA and also at the Department of Computer Science, University of Colorado at Boulder, Boulder, CO, USA
J Riley is at Office of Educational Innovation and Technology, Massachusetts Institute of Technology, Cambridge, MA, USA
BE Granger is at Physics Department, California Polytechnic State University, San Luis Obispo, CA, USA
A Gonzalez is at Department of Computer Science, University of Colorado at Boulder, Boulder, CO, USA
R Knight is at Department of Chemistry and Biochemistry, University of Colorado at Boulder, Boulder, CO, USA and also at the Howard Hughes Medical Institute, Boulder, CO, USA

F Perez is at Helen Wills Neuroscience Institute, University of California at Berkeley, Berkeley, CA, USA and JG Caporaso is at Department of Computer Science, Northern Arizona University, Flagstaff, AZ, USA and also at the Institute for Genomics and Systems Biology, Argonne National Laboratory, Argonne, IL, USA
E-mail: gregcaporaso@gmail.com
[1]These authors contributed equally to this work

References

Afgan E, Chapman B, Jadan M, Franke V, Taylor J. (2012). Using cloud computing infrastructure with CloudBioLinux, CloudMan, and Galaxy. *Curr Protoc Bioinformatics* Chapter 11: Unit11 19.

Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK *et al.* (2010). QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* **7**: 335–336.

DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K *et al.* (2006). Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol* **72**: 5069–5072.

Knight R, Maxwell P, Birmingham A, Carnes J, Caporaso JG, Easton BC *et al.* (2007). PyCogent: a toolkit for making sense from sequence. *Genome Biol* **8**: R171.

Liu Z, Lozupone C, Hamady M, Bushman FD, Knight R. (2007). Short pyrosequencing reads suffice for accurate microbial community analysis. *Nucleic Acids Res* **35**: e120.

Lozupone C, Knight R. (2005). UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microbiol* **71**: 8228–8235.

Pérez F, Granger BE. (2007). IPython: a system for interactive scientific computing. *Computing in Science and Engineering* **9**: 21–29.

Price MN, Dehal PS, Arkin AP. (2010). FastTree 2–approximately maximum-likelihood trees for large alignments. *PLoS One* **5**: e9490.

Stein LD. (2010). The case for cloud computing in genome informatics. *Genome Biol* **11**: 207.

Walters WA, Caporaso JG, Lauber CL, Berg-Lyons D, Fierer N, Knight R. (2011). PrimerProspector: de novo design and taxonomic analysis of barcoded polymerase chain reaction primers. *Bioinformatics* **27**: 1159–1161.

Wang Q, Garrity GM, Tiedje JM, Cole JR. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl Environ Microbiol* **73**: 5261–5267.

Supplementary Information accompanies the paper on The ISME Journal website (http://www.nature.com/ismej)