National Institute of Environmental Health Sciences

Bioinformatics E-Bulletin

a semi-annual communication of cutting-edge bioinformatics/computational biology support entities, scientific software, collaborative research and trends in the field

Jan 21, 2014 Volume 5, Issue 2

In This Issue

Useful hyperlinks

Bioinformatics support experts

BioPlanet

Ontomation

Send comments to:

Editor: Pierre R. Bushel

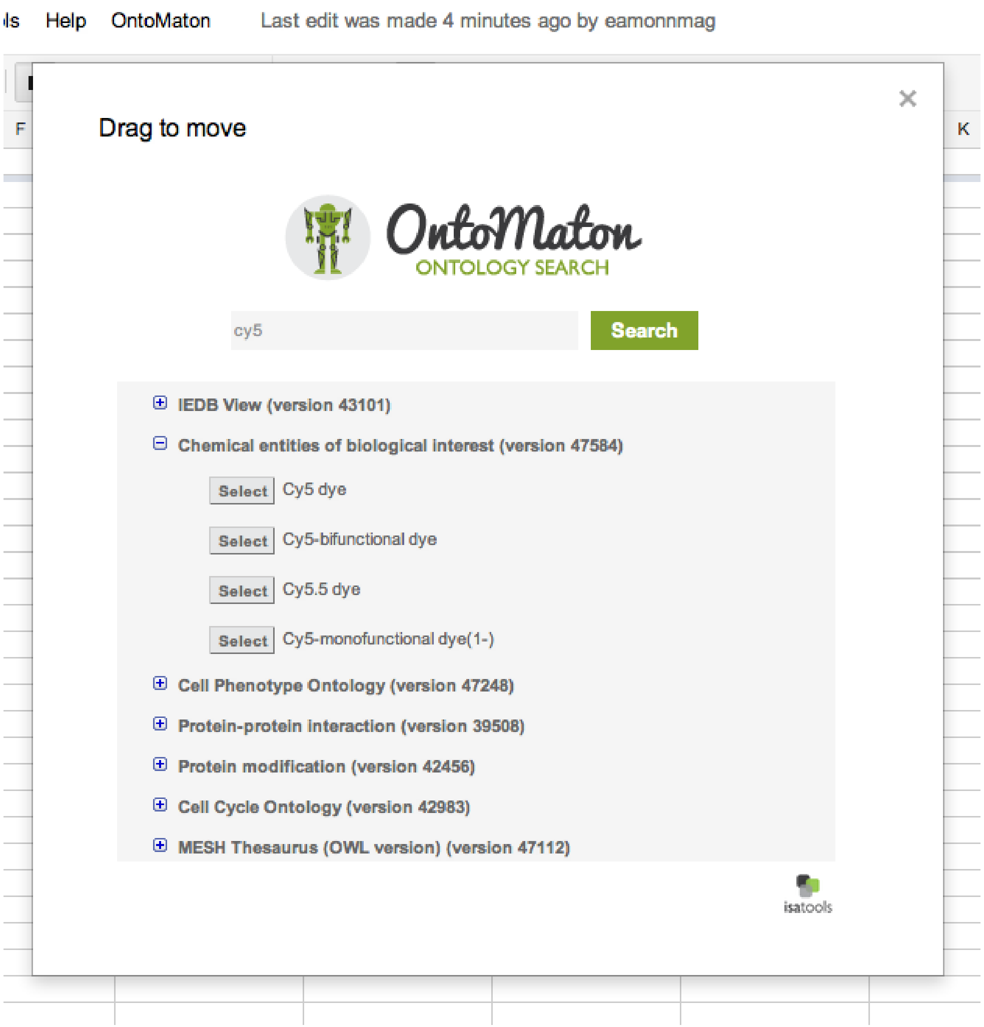
bushel@niehs.nih.gov

Collaborative Data Annotation and Management with Google Spreadsheet and Ontomaton Vocabulary Widget

Contributed by Dr. Philippe Rocca-Serra on behalf of the ISA Team, Oxford e-Research Centre, University of Oxford

Spreadsheets are the staple of data management. They are ubiquitously used to keep track of various experimental data descriptors, from sample names to data file locations and analysis results. However, locally stored documents do not lend themselves very well to twenty first century book-keeping, which, more often than not, requires dealing with collaborators distributed across institutions and continents having to report on specific sections of study. To make matters worse, demands are often placed on scientists to avoid reliance on free text and they favor controlled terminologies to ensure annotation consistency across datasets. Is it possible to deliver on all these fronts? Somehow, yes and the Ontomaton widget allows just that (Maguire et al. 2013). It enables collaborative editing and data reporting while enabling scientists to access vocabulary metadata store - the Bioportal at Stanford University in this instance (Whetzel et al. 2011) - by taking advantage of a freely available, cloud-based spreadsheet environment provided by Google documents and existing web-services. Ontomaton was initially developed to support creation of ISA-Tab records for the European Union FP6 Carcinogenomics integrated program but the widget can be used in any Google document and is independent from the ISA-Tab format.

Installing Ontomaton in a Google spreadsheet is a doodle. Simply open a Google Spreadsheet, go to Menu “Tools”, and select “script gallery”. Searching for “ontology” or “ontomaton” in the script gallery dialog box should return only one hit. Click “install” and follow the instruction to authorize the script to run. Once deployed, the regular Google Spreadsheet menu will be augmented with the “Ontomaton” item. Using the function is straightforward and has 2 modes: simple annotation or tagging.



⊞ IEDB View (version 43101)

⊟ Chemical entities of biological interest (version 47584)

- Select Cy5 dye
- Select Cy5-bifunctional dye
- Select Cy5.5 dye
- Select Cy5-monofunctional dye(1-)

⊞ Cell Phenotype Ontology (version 47248)

⊞ Protein-protein interaction (version 39508)

⊞ Protein modification (version 42456)

⊞ Cell Cycle Ontology (version 42983)

⊞ MESH Thesaurus (OWL version) (version 47112)

By default, one searches the entire content available from Bioportal. However, data managers can work with Ontomaton features to create a customized Google Template with a set of restrictions. These restrictions ensure that only specific vocabularies are accessed in a given field. Finer grained control can be achieved too as, within a given vocabulary, one can also specify which branch to search, thus narrowing down the search space and speeding up the service. Another feature of the widget is the creation of a "terms" worksheet which will log the full details of each term selected for annotation and tagging. These details (a.k.a metadata) include URI, ontology source and version. The rationale for this is simple: it ensures that all the necessary information is available to expose the tabulated records as a resource description framework (RDF) and linked data at a later stage.

Current development consists of bringing Ontomaton up to date with the Bioportal latest application programming interface (API) release (version 4.0) and additional work is underway to expand beyond the Bioportal vocabulary service to include access to the Linked Open Vocabulary registry in order to further support scientists and help them expose their data in the semantic web.

One word of caution: as with all cloud-based solutions, one needs to be aware of legal implications associated with virtual infrastructures as they may conflict with local, institutional guidelines. One may therefore have to seek approval prior to using google documents for tracking research metadata.

Having said that, the work highlights the cost effectiveness of such an approach as is demonstrated by the functionality and productivity gains that are obtained with minimal investment.

OntoMaton: a bioportal powered ontology widget for Google Spreadsheets.
Maguire E, González-Beltrán A, Whetzel PL, Sansone SA, Rocca-Serra P.
Bioinformatics. 2013 Feb 15;29(4):525-7. doi: 10.1093/bioinformatics/bts718. Epub 2012 Dec 24.

BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications.
Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, Musen MA.
Nucleic Acids Res. 2011 Jul;39(Web Server issue):W541-5. doi: 10.1093/nar/gkr469. Epub 2011 Jun 14.

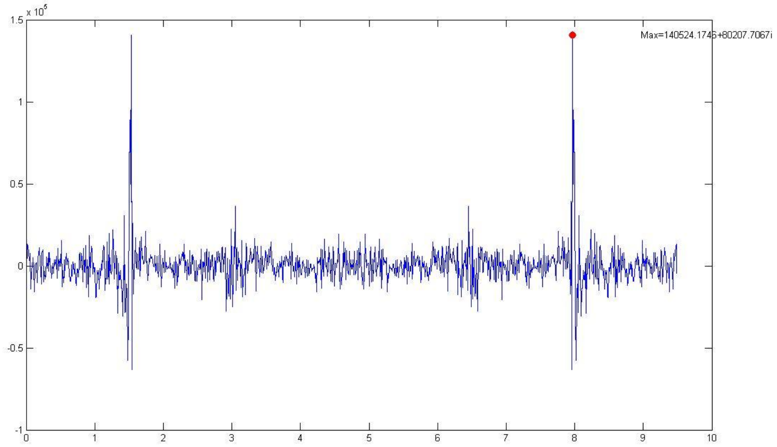
Programming in Matlab: *Not as hard as you may think*

Contributed by Mr. Jeff Tucker

I took my first college level computer programming class 20 years ago while a sophomore at NC State. Back then you either learned Pascal or FORTRAN and your major really dictated which one. While in undergraduate and graduate school I was exposed to several programming languages (Pascal, Basic, C++, Fortran, MC6800 assembly, HTML, java, etc.) but honestly I never mastered any of them. I love electronic hardware and optics; software and programming just isn't for me. However, the one thing I have learned about programming is that regardless of language, the basics are really the same. In almost all programming languages you are dealing with variables, Boolean conditions, branching, and input/output. The syntax varies but the basic rules are the same.

A few months ago, I had a user of the Fluorescence Microscopy and Imaging Center (FMIC) approach me regarding an idea they had for part of their K99 grant application. They ultimately wanted to measure the frequency of beating cilia and quantitate how one specimen varies from another. It should come as no surprise that cilia are very small and combine this with the fact that they beat at a rate of 20 times per seconds, and you will find that they are also rather hard to image. However, we determined that if you use a high speed camera capable of imaging at 100 frames per second that we could capture a multiple, small regions of interest (ROI) that contained cilia. The beating cilia caused changes in image intensity in the ROIs and when we plotted this intensity data over time, we could see an oscillation! The next step in the analysis was to figure out how to measure this oscillation. I recalled from graduate school that any time signal is composed of a series of frequencies and that an easy way to determine the frequency components of a time signal is to do a Fast Fourier Transform (FFT). While this sounds like an easy solution, I quickly realized that I didn't know how to perform the FFT operation. After a few google searches I realized that most people do this type of conversion with Matlab and since NIEHS has licenses for Matlab, I figured that was the best path forward.

After installing Matlab and opening it up I was immediately flashed back to graduate school because I was staring at a command line which was waiting for me to type in some sort of programming syntax. Fortunately my minimal programming experience reminded me that Matlab is just another programming language and that all I really need to do is learn the syntax and then use the help files to guide me through the process of entering my data, running a FFT, and plotting it out. At that time I also decided to contact Dr. Pierre Bushel in the Biostatistics Branch in hopes of shortening my learning curve. Pierre came over to the FMIC and he gave me a quick overview of Matlab. He was able to answer a few questions and after spending just a few minutes with him, I was able to navigate Matlab enough to load my data, run a FFT, and then plot it out!



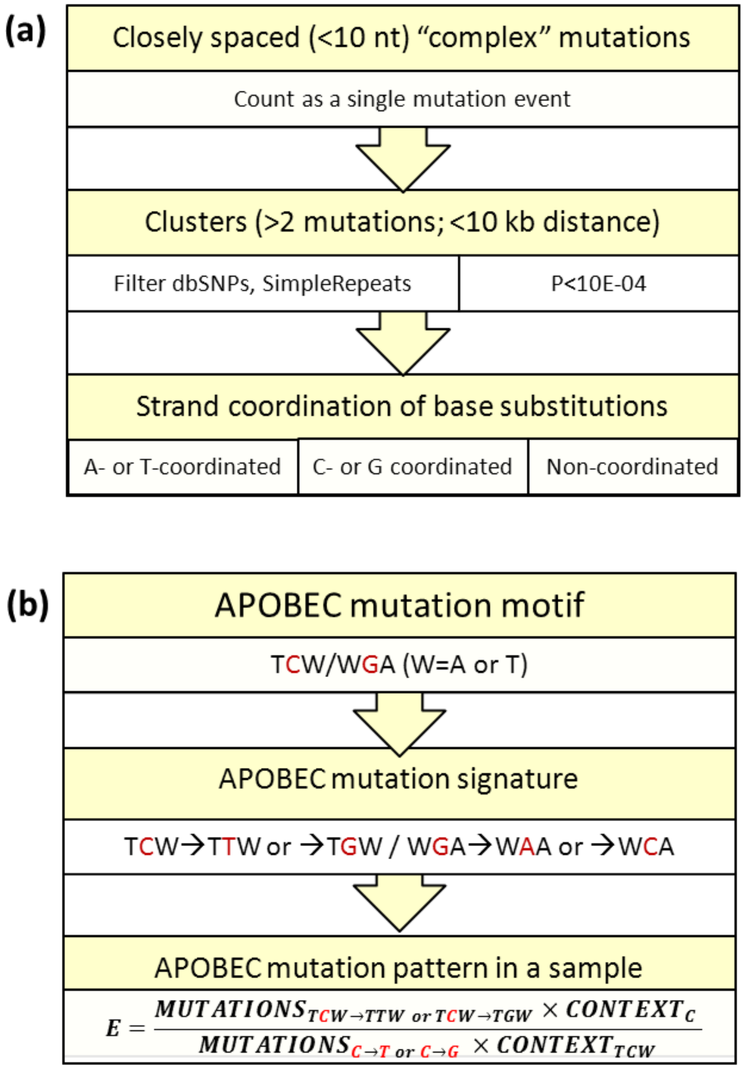
In the end, I was surprised how easy it was for someone with dusty programming skills (at best) was able to achieve quality results with Matlab. Next time your research begins to become computationally intensive, keep in mind the resources/solutions NIEHS has in terms of Matlab and the helpful staff at the Biostatistics Branch and in the Bioinformatics group.

Building a User-configurable Data Processing Pipeline

Contributed by Dr. Les Klimczak

While bioinformaticians are usually very successful in conducting their own analyses to answer questions posed by biologists, it is often a challenge to scale up and automate the analytical work performed by biologists themselves. Such work involves most often very fragmented ad hoc operations in multiple poorly structured Excel files that do not lend themselves easily to standardization and automation. Our collaboration with Drs. Steven Roberts and Dmitry Gordenin from the Chromosome Stability Group represents a rare example of a successful migration from individual analytical operations developed step-by-step and bottom-up by the biologists themselves in Excel to a fully automated "single-click" multistep pipeline operating on multiple large files and implemented in R.

The pipeline performs analyses of sequence motifs in the vicinity of mutation calls collected in Mutation Annotation Format (MAF) files generated by Next Generation Sequencing (exome or whole genome) projects and, while it could be generalized to any type of mutation in any type of data, it focuses on detecting the overrepresentation of the APOBEC mutation signature in tumor samples (see Figure).


(a)
Closely spaced (<10 nt) "complex" mutations
Count as a single mutation event
Clusters (>2 mutations; <10 kb distance)
Filter dbSNPs, SimpleRepeats
P<10E-04
Strand coordination of base substitutions
A- or T-coordinated
C- or G coordinated
Non-coordinated
(b)
APOBEC mutation motif
TCW/WGA (W=A or T)
APOBEC mutation signature
TCW→TTW or →TGW / WGA→WAA or →WCA
APOBEC mutation pattern in a sample
$E = \frac{MUTATIONS_{TCW \to TTW \text{ or } TCW \to TGW} \times CONTEXT_{C}}{MUTATIONS_{C \to T \text{ or } C \to G} \times CONTEXT_{TCW}}$

The pipeline operates in the following steps:

- it uses the coordinates of the called mutations to retrieve the +/- 20 bases context from the genome sequence and to count the presence of given sequence motifs,
- identifies those mutations that are spaced close to each other (clustered),
- generates various per sample summaries for each sequence motif,
- and tests for their overrepresentation.

The operations were developed bottom-up by the biologists in Excel and were later generalized and scaled up in R scripts corresponding to the individual steps and coordinated by a controller script written in R as well. This required a disciplined focus on standardizing the data sets and operations that is rarely achievable in a biological lab. Furthermore, it relied on fruitful interactions between the biologists and the bioinformaticians. In addition to a number of simple parameters that can be provided to the pipeline using a GUI interface or configuration files, there are several complex modifications that have been designed to be defined by the biologists and to be easily "pluggable" as code modules without major rewriting of the scripts.

The list of sequence motifs to be searched for is provided as two R vectors of the motif IUPAC names and their corresponding regular expressions (for instance: "tCw" and "tC[at]") can easily be constructed by biologists as well as debugged by them with only minimal expertise in R! The IUPAC names of the motifs are used as names of the columns in their summary counts and as variables representing those counts in formulas that again can be easily modified and proofread by biologists (for instance: (c + g) - (tcw + wga)). The rules for creating per sample summaries can be generated on-the-fly from entries in tab-delimited files specifying the names of MAF columns to be considered for inclusion of the counts and their values.

The application of the pipeline to detect the overrepresentation of the APOBEC signature in various tumors has been published (Roberts et al. 2013) with the pipeline development team including Les Klimczak, Arpit Tandon, Depak Mav, Shawn Harris, Ruchir Shah and David Fargo and we are currently collaborating with several cancer consortia to include the pipeline results in their analyses as well as working with the bioinformatics group of the Broad Institute to incorporate the pipeline into their Firehose platform.

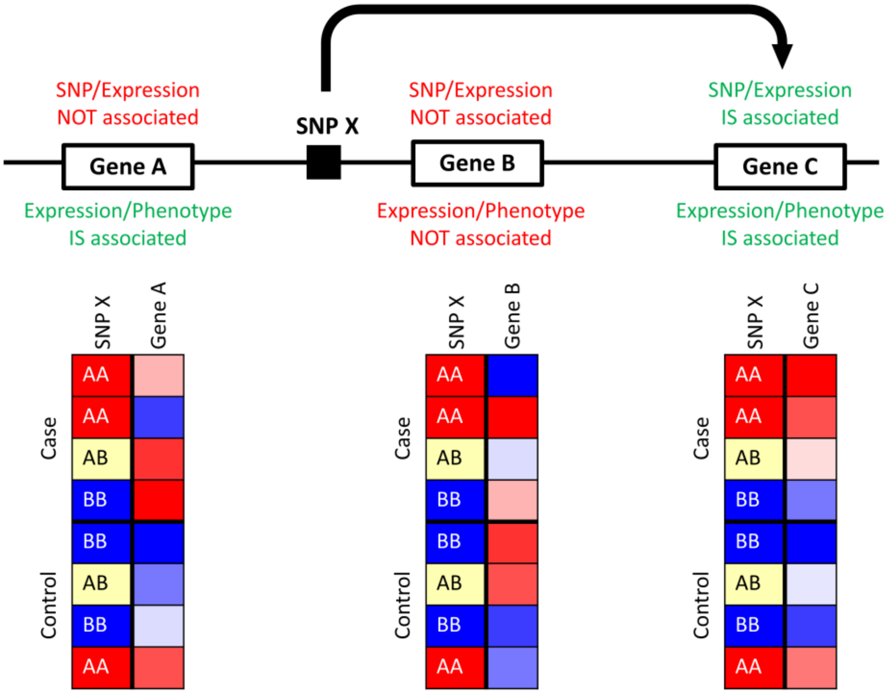
Roberts, SA, Lawrence, MS, Klimczak, LJ, Frage, D, Stojanov, P, Kiezun, A, Kryukov, GV, Carter, SL, Saksena, G, Harris, S, Shah, R, Resnick, MA, Getz, G, Gordenin, DA. An apobec cytidine deaminase mutagenesis pattern is ubiquitous in human cancers. Nature Genetics 45: 970-976, 2013.

Integration of Gene Expression and Genotype Data using an eSNP Analysis

Contributed by Dr. Brian Bennett

Researchers have performed countless genome-wide association (GWA) studies attempting to identify single-nucleotide polymorphisms (SNPs) associated with a phenotype of interest. However, it is often unclear how these SNPs are impacting the phenotype or which genes these SNPs are influencing. An eSNP is a SNP/gene pair where changes in the SNP are associated with expression differences in the gene. An eSNP analysis may lead to the identification of instances where changes at a genotypic locus are driving gene expression differences that influence a given phenotype.

For example, for a SNP that is associated with a given phenotype, SNP X (see Figure), it would be beneficial to know which gene(s) the SNP is influencing and affecting the phenotype. One common method for determining the most probable gene is to select the gene closest to the SNP, Gene B (see Figure). However, this may not be the gene influenced by the SNP and driving phenotypic differences. It would be useful to identify genes whose expression is both associated with the SNP and with the phenotype, like Gene C (see Figure).


SNP/Expression NOT associated
SNP X
SNP/Expression NOT associated
SNP/Expression IS associated
Gene A
Gene B
Gene C
Expression/Phenotype IS associated
Expression/Phenotype NOT associated
Expression/Phenotype IS associated
SNP X
Gene A
Case
Control
AA
AA
AB
BB
BB
AB
BB
AA
SNP X
Gene B
Case
Control
AA
AA
AB
BB
BB
AB
BB
AA
SNP X
Gene C
Case
Control
AA
AA
AB
BB
BB
AB
BB
AA

This eSNP analysis method embodies principles of expression quantitative trait loci (eQTLs) and is based on a method developed by Schadt et al. (2008). The first step is to assemble a list of phenotypically associated SNPs, using any suitable method. The next step is to identify a list of target genes for each phenotypically associated SNP. This can be accomplished by performing an eQTL analysis and defining the set of genes whose expression is associated with each phenotypically associated SNP as the initial list of gene targets. The final step is to filter this list to only include genes whose expression is associated with the phenotype, using any suitable method. This will result in a final list of SNP/gene pairs where the SNP is associated with the phenotype, the SNP is associated with the expression of the gene, and the expression of the gene is associated with the phenotype. This list provides possible instances where variation at a genotypic locus may drive gene expression changes that influence phenotypic differences.

There are many ways in which SNP changes can influence phenotypic differences. This method is slightly restrictive in that it will not identify instances where genetic variations in SNPs are influencing a phenotype by mechanisms other than changing the expression of genes. For example, this method will not identify a SNP in the protein-coding region of a gene that alters the structure of the protein produced by the gene but does not alter the expression of that gene. However, SNP/gene pairs that are identified by this method may provide additional insight into the underlying biology driving the phenotype and can be a powerful way to identify potential gene targets for future research.

Schadt EE, Molony C, Chudin E, Hao K, Yang X, et al. Mapping the genetic architecture of gene expression in human liver. PLoS Biol. 2008;6(5):e107.

Paralleling R Processes

Contributed by Mr. Jianying Li

R has been widely accepted and adopted as a formal statistical programming language since it became alive in the late 90s. From then on, it has played a pivotal role in major battlefields when statistical analysis is performed. With the advance of modern biological technology i.e. the next generating sequencing, scientists are deemed to face ever-growing scientific data. This not only completely reshapes biologists' mindset, unfortunately it also presents challenges to the statisticians/bioinformaticians, who often need to defend themselves over extended period of waiting time, which is often time largely determined by the unmatched computational resources

Leveraging the parallel capability becomes a natural solution for huge datasets. Other than the GPU concept, which I addressed in an earlier NIEHS Bioinformatics E-Bulletin (vol. 5, issue 1), another solution is to utilize the multiple cores equipped with modern-designed processors. Since R is not designed for easy parallel programming, usually it will take the experienced programmers and special skills to modify the detail implementation of major algorithms to achieve such a goal. In 2008, a group of researchers from Spain introduced an R/parallel package (Vera, Jansen et al. 2008), which could be integrated with any existing R packages, it provided a user-friendly framework without any modification of the existing code and ensured the analysis ran in a parallel mode. According to their assessment, with this R/parallel package, the processing time could be reduced by N-fold where N is the number of available processor cores. Unfortunately, it seems that this team has stopped further development of this package. The existing package has become incompatible with updated versions of R and the embedded code stopped working properly.

As stated above, the need for paralleling R processes never stops. On the contrary, it becomes more urgent to have a workable solution. As a result, all the attention has been drawn to another robust package called doMC (MC stands for multi-core) which is developed and maintained by Revolution Analytics (http://www.revolutionanalytics.com/). Currently, the doMC package has been widely used in the statistical and computational analysis communities. A recent research effort conducted by Dr. Liang Niu at the NIEHS Biostatistics Branch and geared towards testing the differential isoform usage from the RNAseq data largely relied on the parallel computing. He adopted a doMC strategy to handle the parallel computing, which significantly improve the performance. Dr. Niu published his R package called IUTA (Niu 2013) in December 2013 with a required dependency on doMC. In another personal communication with an individual at NIEHS who desired parallel processing in R, I learned that there were difficulties trying to use R/parallel to adapt an analysis algorithm in parallel mode. It became apparent that the R/parallel package stopped working but the newly available doMC package worked quite well.

In order to demonstrate the improved performance, the following example code (test_doMC.R) inherited from the doMC package was implemented:

```
library(doMC)
registerDoMC()

x <- iris[which(iris[,5] != "setosa"), c(1,5)]
trials <- 10000
```

```
ptime <- system.time({
  r <- foreach(icount(trials), .combine=cbind)
%dopar% {
    ind <- sample(100, 100, replace=TRUE)
    result1 <- glm(x[ind,2]~x[ind,1],
family=binomial(logit))
    coefficients(result1)
  }
})[3]

cat(sprintf('Parallel time using doMC on %d
workers: %f\n',
            getDoParWorkers(), ptime))

stime <- system.time({
  r <- foreach(icount(trials), .combine=cbind)
%do% {
    ind <- sample(100, 100, replace=TRUE)
    result1 <- glm(x[ind,2]~x[ind,1],
family=binomial(logit))
    coefficients(result1)
  }
})[3]

cat(sprintf('Sequential time: %f\n', stime))
cat(sprintf('Speed up for %d workers: %f\n',
            getDoParWorkers(), round(stime /

ptime, digits=2)))
```

The following shows the running log and the analysis results between running sequentially and using multi-cores on one of NIEHS' high-performance computing servers. The parallel run with the doMC package greatly improved the performance (speed up by 8.56 times!)

```
> getwd()
[1] "/ddn/gs1/home/li11/project2014/MGI-eBulletin"
> source ("test_doMC.R")
Loading required package: foreach
foreach: simple, scalable parallel programming from Revolution
Analytics
Use Revolution R for scalability, fault tolerance and more.
http://www.revolutionanalytics.com
Loading required package: iterators
Loading required package: parallel
Parallel time using doMC on 24 workers: 7.860000
Sequential time: 67.247000
```

```
Speed up for 24 workers: 8.560000
```

Although doMC has provided a solution for paralleling R processes, there are a few caveats one must be aware of. As of the time this note is being drafted, (1) doMC only works with the Linux operating system and Mac OS. It does not support R under Windows. (2) by design, doMC will register processor cores before executing the jobs, it looks for the available cores but only registers half of them. (3) It needs a little more refining of the code (comparing to R/parallel) to create a special “foreach” loop. (4) Lastly, it does take extra time to distribute the jobs to the registered cores. In one testing case, doMC ran slower than the sequential process when dealing with a simple algorithm (data not shown).

In conclusion, referring back the NIEHS Bioinformatics E-Bulletin topic about BigData (vol. 5, issue 1), Revolution Analytics has introduced the “rmr” package, which allows researchers to conduct large-scale statistical analysis via a MapReduce framework on a Hadoop cluster. It is another level of the “paralleling” concept in a sense that the processes will be reduced to smaller chunks and run on separate clusters and accessing data via a novel “hdfs” file system. Within the MapReduce framework, it follows a well-defined map-reduce framework in terms of implementing the algorithms. In the meantime, it opens up a much wider window for the BigData problem with one limitation being the number of cluster nodes. Stay tune for more discussion in a future issue of the NIEHS Bioinformatics E-Bulletin.

Li, J. (2013). "Leveraging the RGPU for BigData analysis." NIEHS Bioinformatics E-Bulletin 5(1).

Niu, L. (2013). "An R package -- IUTA."

Vera, G., R. C. Jansen, et al. (2008). "R/parallel--speeding up bioinformatics analysis with R." BMC bioinformatics 9: 390.

Bioinformatics By the Way

<u>Winter 2014 Introduction to Biostatistics and Bioinformatics Short Courses</u>

Presented by members of the Biostatistics Branch and Integrative Bioinformatics

NOTE: All courses will be presented in the NIEHS Rall Building 101 A012 Training Facility. A012 is limited to 24 registrants. The Training Facility is outfitted with Macintosh and Windows laptops. For any course that is hands-on, you will be provided a ready-to-use laptop.

Registration for NIEHS staff will open approximately 2 weeks before the course date.

If space remains available at 1 week before the course registration will open to non-NIEHS staff. Non-NIEHS staff are required to present a valid government issued ID when entering our campus. You will be contacted about the process as part of your registration confirmation.

Course handouts (PowerPoint slides, sample data, etc.) are typically available as part of each course description a few days before the presentation date.

Please contact Bill (qb) Quattlebaum (quattleb@niehs.nih.gov) if you have questions.

See the following web site for information about registering and for course materials:

http://junction.niehs.nih.gov/divisions/dir/resources/analysis/ib/courses/index.htm

1. Introduction to statistics and experimental design

Presented by Casey Jelsema

Thursday, 23-January, 1:30 PM – 3:30 PM

AND Friday, 24-January, 1:30 PM – 3:30 PM

NOTE: This is a 2-part course. You will be registered for and must plan to attend both parts of this course. Please do not sign up if you can only attend one part.

Abstract: This class provides an introduction to statistics and experimental design. Topics include experimental design, levels of measurement, numerical and graphical summarization of data, and sample size determination. The principles of statistical inference are also discussed, including confidence intervals and hypothesis testing. Examples are used throughout to illustrate the utility of many of the methods.

Please email a registration request to Bill (qb) Quattlebaum

2. Testing statistical hypotheses

Presented by Alison Wise

Tuesday, 18-February, 9:30 AM - 11:30 AM

Abstract: An approximate list of topics: Formulation of null and alternative hypotheses. Power and sample size calculations. Some common parametric and nonparametric tests. Resampling based methods: Permutation tests, bootstrap methodology. Basic principles of multiple hypothesis testing: 1 Two common notions of false positive rates - family wise error rate (FWER) and false discovery rate (FDR). 2 Common methods for controlling FWER and FDR.

Please email a registration request to Bill (qb) Quattlebaum

3. DNA Microarray Analysis

Presented by Keith Shockley

Friday, 21-February, 1:30 PM – 3:30 PM

Abstract: This introduction to DNA microarray data analysis will begin with an overview of microarray technology and the quality control and normalization of microarray data. Data preprocessing, experimental design and gene expression clustering will be discussed. Analysis of variance (ANOVA) will be introduced to compare two or more treatment groups. The class will incorporate numerous examples and include references to widely available commercial and open-source software.

Please email a registration request to Bill (qb) Quattlebaum

4. Introduction to NGS tech and ChIP-seq

Presented by James Ward

Tuesday, 4-March, 9:30 AM – 11:30 AM

Abstract: We will start by looking at the evolution of sequencing technology and briefly go over the chemistry and principles behind first, second, and third generation sequencers. We will then delve into the methods and applications for ChIP-seq analysis. ChIP-Seq stands for Chromatin Immunoprecipitation followed by Sequencing, and as the name suggests, uses immunoprecipitation to assess the genomic loci of DNA binding proteins such as transcription factors, histone modifications, CTCF etc. This section will include an overview of the ChIP-seq protocol, experimental requirements, statistical analyses, data formats and visualization. We will end with a demonstration of ChIP-seq data analysis, starting with raw data from the sequencer to the visualization of “Chipped” peaks in the UCSC genome browser. Please note that due to time limitations, we will not be able to perform a real-time demonstration and will instead rely on a recently analyzed example. Bringing a laptop is not necessary.

5. Pathways analysis

Presented by Jianying Li

Friday, 7-March, 9:30 AM – 11:30 AM

Your choice of Macintosh or Windows laptops will be provided.

Abstract: This short course will provide an overview of a few of the current knowledge-based pathway analysis approaches widely used in genomic research. It covers basic information on frequently used databases (i.e. gene ontology categories), as well as curated knowledge bases from scientific literature and the public domain. It also focuses on two commonly used statistical approaches

(Hypergeometric/Fisher exact test and Kolmogorov Smirnov) in pathway analysis with a hint of theoretical illustration. In this course, we will introduce a few (publicly available/free access and license-based) applications which implement either of the two statistical approaches. In the end, we will touch on some questions about the pros and cons in the pathway analysis package(s) and explore ways to deal with such concerns. The goal is to provide the scientists at the institute with a solid understanding of the fundamental concepts behind commonly used analytical tools for pathway analysis in order to empower them to maximize the analysis of their data and enhance interpretation of their results.

6. Introduction to UNIX and Perl for Biologists

Presented by Adam Burkholder

Tuesday, 18-March, 1:30 PM – 4:30 PM

NOTE: This is a 2-hour course but with an additional hour for questions and hands-on UNIX with Adam.

Abstract: This course is intended as an introduction to the UNIX computing platform for biologists with no prior experience. It will include an overview of the scientific computing resources available at NIEHS, and cover the most commonly used and most useful built-in UNIX commands. Additionally, the Perl programming language will be introduced in the form of easy-to-use "one-liners". This course will consist primarily of live demonstrations using Mac OS X, so attendees with access to a Macintosh laptop are encouraged to bring one.