






Posts

Recent Tegu genome paper serves as a great primer for non-model genome assembly

January 08 2019

Every once in a while one comes across a particularly relavent paper - one that backs up something you have been suggesting in theory, one that has great examples of analyses, one that has particularly great discussion, or even one on particular favorite organism. These papers are great to come across when you do a bunch of educational outreach as we do (ahem...), as we can use them to illustrate methods, use the data as demos, etc. Recently, I ran across one that checks off ALL of these boxes and wanted to share thoughts here on our blog, couched in the topic of "so you are thinking about doing a genome project"!

The paper is the recent tegu genome paper (linked here). This paper struck

me for the following reasons:

1) I come from a herpetology background and I currently live with an adorable Tegu.
2) The paper follows almost exactly what we have been recommending to people for genome projects as far as sequencing
3) It does a great job of explaining and listing the sheer volume of bioinformatic packages needed to do a genome
4) I disagree with their choice of transcriptome analyses, but only in one particular point.

1) Tegus are amazing.

Tegus are amazing creatures - and quite interesting pets. The reason the authors selected the tegu was because lacertids were missing from the sequenced reptile genomes, a nice addition for comparative whole-genome work in reptiles. They are also in the pet trade (making them easy to work with in future research) and of economic importance to their native Southe American range where they are used for leather and meat production.

They also have pretty facinating biology. A fairly recent finding about their ability to be endothermic during breeding season can be found here (scientific paper linked at the bottom). I look forward to some genomic follow up on this work - with a genome, transcriptome, and variable endothermy, it

Research Desktop (RED) for new users

Manipulating Tar Files

Debugging job errors

Running Anvio on Jetstream

Adding Users to Jetstream

Installing conda packages locally

Third Generation Sequencing Technologies

would be relatively easy to design a study to investigate the gene regulation changes associated with the change!


Moira enjoying the sun

And since everyone needs more cute lizard pictures - here is Moira, basking in the sun like the dinocat she is ^_^.

for Microbiome analysis

Getting Started with Ansible

Setting up Volumes on Jetstream

Setting up Globus on Jetstream

Workflow Management

Getting started on Bridges

NCGAS now has resource allocation specifically for workshop

2) Recommended Genome Assembly and Analysis

Since I delved into learning everything I could about the different Third Generation Sequencing Topics (see our blog here!), I have been suggesting the following as a general starting plan for genome projects on non-model critters:

High Accuracy Short Reads: 30-50X Illumina
Less Accurate Long Reads : 30-50X PacBio
Some form of Optical Map : Bionano*

*HiC can go here too, but I personally suggest Bionano for genome assembly projects because it is cheaper - that said, HiC is a great tech and has many additional uses that make is super appropriate for certain cases or downstream analysis. More on that below.

This isn't anything revolutionary, it is more or less the blueprint of what many genomes are doing these days, especially non-model organisms or projects with less funding, or previous sequencing.

The tegu falls under two (possibly three) of those categories - it is not a model system, and it did have a previous Illumina only genome assembly, which the authors built upon with more Illumina, PacBio, and (you guessed it) Bionano. As such, it serves as a great example of how to approach this kind of project.

support for clients!

Running Canu on Carbonate

NCGAS has now added XSEDE resources to our field station support!

Join NCGAS at PAG XXVI!

Command Line Shortcuts

Getting Started on Globus

Getting Started on Jetstream

First, let's look at the data used by the paper:

Tech	Source	Coverage	Reads/Details	Genome Version
Illumina	Liver from one male tegu	41X	2x300bp MiSeq	v1, v2
		33X	2x150bp mate-pair (2 x 2Kb mate-pair libs, 2 x 10Kb mate-pair libs - HiSeq 2500)	v1, v2
PacBio	Liver from one male tegu	29.8X	10ug gDNA sheared to 10-25kb, selected for 9Kb+, 10ug of gDNA sheared to 40Kb, selected for 10Kb+, run on 205 SMRT cells (PacBio RSII)	v2
Bionano	Embryonic tissue	N/A	IrysPrep v1.1.12, PFGE selected for 100Kb-1Mb in length, run on 5 flow cells	v2
Illumina - Transcriptome	Embryonic tissue	N/A	2x75bp, 8 strand specific mRNA libraries, HiSeq 2500	v2

The ultimate result - a genome with a scaffold N50 of 55.4Mb, a total size of 2.068Gb (~2Gb is estimated size of Tegu genome), and some of the best metrics in reptile genomes.

As you can see, this high quality, highly contiguous genome is the result of almost exactly what we recommend. **But why do we recommend this?** The concept here is that there is a trade off between having high accuracy and getting reads long enough to scaffold your data into contiguous regions. So we go with an approach that implements **successive approximation**. Let's look at this a in reverse of how it is usually done.

Bionano has great long range architecture information, but by nature of it's technology (see here for a review), only 1Kb resolution. This alone is not terribly useful, but it serves as a very hazy blueprint for the genome - the structure is there, but to get more resolution, more data is needed.

This is where read technologies come in. Long-read technology is growing in popularity (as it gets cheaper and more accurate), but PacBio still seems to dominate for general use analyses like sequencing a genome (for a review of this tech, see here). Long-read technology is still a bit less accurate and more expensive that short-reads, but it increases the resolution of Bionano, giving you single base resolution, but with more errors than you might want.

are making more genomes feasible

Third Generation Sequencing

Job Submission Guide

Data Management

ScienceNode: Rattlesnakes on Jetstream

Publication: Dragon blood and monster spit

Analysis: A Guide to Tree

High accuracy is where Illumina short-reads shine. Still more accurate and cheaper than PacBio, Illumina allows you to correct any error prone points in the PacBio assembly. You could forego Illumina and really invest in PacBio reads to boost accuracy, but this likely isn't quite cost effective yet. You get more coverage at a higher accuracy with Illumina, so it's still generally recommended.

However, do note you cannot skimp too heavily on the PacBio in lieu of high coverage of Illumina. This is actually discussed in the reviews of the paper (which are blessedly included!):

"[T]he safest way of using PacBio long reads technology is to produce high depth of data and polish the raw reads by self-correction for systemic error. Although the short reads correction could correct the mismatch, the method has limit capacity in dealing with the indel errors for pacbio reads."

This is critical! Do not be tempted to ignore idel errors that are hard to correct without self-correction! However, there is hope if you don't have the cash for a ton of PacBio data, as the authors of the paper point out:

"Proovread is one of the most accurate methods and we specifically used it because it does not break the longer-than-usual Illumina MiSeq reads into smaller kmers for a De-Bruijin graph based error correction... Proovread was computationally very expensive, we found that this method and our data is

Building

Archiving 101

NCGAS Blog Begins!

able to correct indel errors in PacBio reads."

They also point out that they would see signatures of errors in their downstream analyses:

"[I]t is expected that uncorrected base errors (in particular uncorrected indels) would create frameshifts in genes, with BUSCO analyses would detect as fragmented genes".

Do note that the longer-than-usual MiSeq reads were what was important here - be careful if you want to apply this to standard length Illumina HiSeq!

Also, please note: These suggestions are just a start point in design!

The point here is that with transcriptomes, we recommend using multiple technologies to overcome some technological biases that may be present (see here for our discussion on assembling transcriptomes). These three suggested inputs are just a start point - there may be reasons to use 10X, HiC, etc for the first two steps - see here for pros/cons/costs and which techs work well together. For example, it might make sense to forgo some PacBio in lieu of heavy coverage in Illumina if you plan to do something like population level analyses directly after. It might make sense to use HiC instead of Bionano when working with samples with high risk of contamination. More on this after PAG!

That said, I would really love to see the end of the step-wise genome. So many projects (such as this one) try to do an Illumina only assembly on a large genome, go through the assembly, clean up, etc., only to find that they need to add other technologies to the mix and REanalyze everything. This takes a lot of time, and likely leads to a higher level of Illumina coverage needed in the end. I personally believe it is time to just accept the fact that you will likely need to include a mix to get a high quality genome.

Just to drive home this point, let's look at the stats in the two versions of the genome - v1: Illumina only, and v2: all three technologies recommended:

	v1	**v2**
N50 scaffold	28.1Mb	55.4Mb
N50 contig	175.8Kb	512Kb
# scaffold	5,988	4,513
# contig	36,428	18,096
total size	2.026Gb	2.068Gb

The improvement is significant, but really most of this is likely due to the

PacBio and the Bionano, not the almost 80X Illumina coverage. They could have gotten away with a lot less if they had planned to do all three from the start - a note to all trying to be conservative with cash!

3) The hard and long part - analysis

This paper had great coverage of how they built their genome, despite having the above pitfall of trying to do an Illumina-only assembly first. Their methods are pretty easy to follow and I don't have a ton to comment on, as we are still determining what we want to recommend here.

However, just looking through my notes on methods, I count 23 different software packages (two of which are large pipelines comprised of several packages e.g. PASA) just to **assemble** the genome/transcriptome. That number jumps to **more than 44 packages needed** when you include downstream analysis such as annotation.

Lesson here: BIOINFORMATICS IS GOING TO TAKE A LONG TIME AND YOU NEED TO BUDGET FOR THAT

Many, many times we hear "you can do a genome for $10K" or whatever number. This does NOT include person time to do the analyses, which are numerous. This does not include computational time, which can be expensive if you don't plan accordingly. While the data might be increasingly cheap to

generate, the unmentioned costs are still high. Planning here for budget as well as what you plan on doing from the get go will only help keep these costs from ballooning (e.g. not having to figure out six different software packages that are all dead ends, while still paying your grad student for the time!).

NOTE: This is the part we at NCGAS are striving to help with - training people to do this kind of thing, providing software support to save you from tearing your hair out trying to install 44 packages, and providing or pointing you toward machines capable of completing the analyses.

There are three other major bioinformatic points I'd like to make about this paper:

- Downstream examples: This paper has great examples of how to do quality assurance, annotation, and some interesting first investigations into the genome you just produced. These are worth looking into and trying with your data, as this paper didn't go far into the weeds of species specific questions.
- Software reporting: If you are going to approach a genome project, please pay attention to how software is reported here - versions and parameters are reported for replicability! This is near impossible to dig back up if you aren't thinking about it upfront - so keep this in mind and keep a log as you go! Versions update on machines and often it can be VERY unclear what version you used for your analyses!

- Browsers: If you are going to do a genome project, expect to provide a browser. This can be a bit cumbersome, but we do have free resources for setting up and running a browser server (see here for more details)!

4) A note about transcriptomes

While I really appreciated this paper for many reasons, the one thing I wish would have been done differently is the transcriptome assembly. I know this wasn't the focus, and what they did was completely legitimate, I just feel like one computational step further would have improved things.

If you are familiar with NCGAS, you probably know we are largely a transcriptome shop - mainly because this has been where new people jump into genomics and HPC use. You are also probably familiar with our transcriptome pipeline (link to the summary and github), and our incessant talk about using multiple assemblers and parameters (see our talk here), just like we talked about above in genome assembly. We often talk about this in the context on not having a genome to work with - but I believe *de novo* assembly is just as important when working with a genome project.

This paper used only genome-based assembly methods for their transcriptome. This was done with Tuxedo software (HISAT2 and Cufflinks) as well as Trinity's genome guided version (which is partly *de novo*). While this is great when you have a genome, I wish they would have done a proper *de novo*

assembly of the transcriptome. *de novo* assembly assumes nothing from the genome and will often find novel transcripts missed in the genome assembly - giving both a quality metric for the genome (% genes transcribed that are not on contigs) as well as a higher quality tool for future analyses.

We asked the authors, and their rationale for foregoing *de novo* transcriptome analysis was to avoid misassemblies that can be a problem with Trinity. We agree this is true - which is why we at NCGAS recommend using several assemblers. I may run their data (the glory of public data repos) and see if there is much added, for my own curiousity.

Conclusion

We will be developing more materials on genome assembly and annotation/analysis in the near future (PAG next week!), but this is a great starting point for those of you considering projects such as these!

« Join NCGAS at PAG XXVII!

NCGAS Returns from PAG Victorious »

Photo credit: Header visualization of miRNA in diffuse large B-cell lymphoma is property of Martin Krzywinski and used with permission.