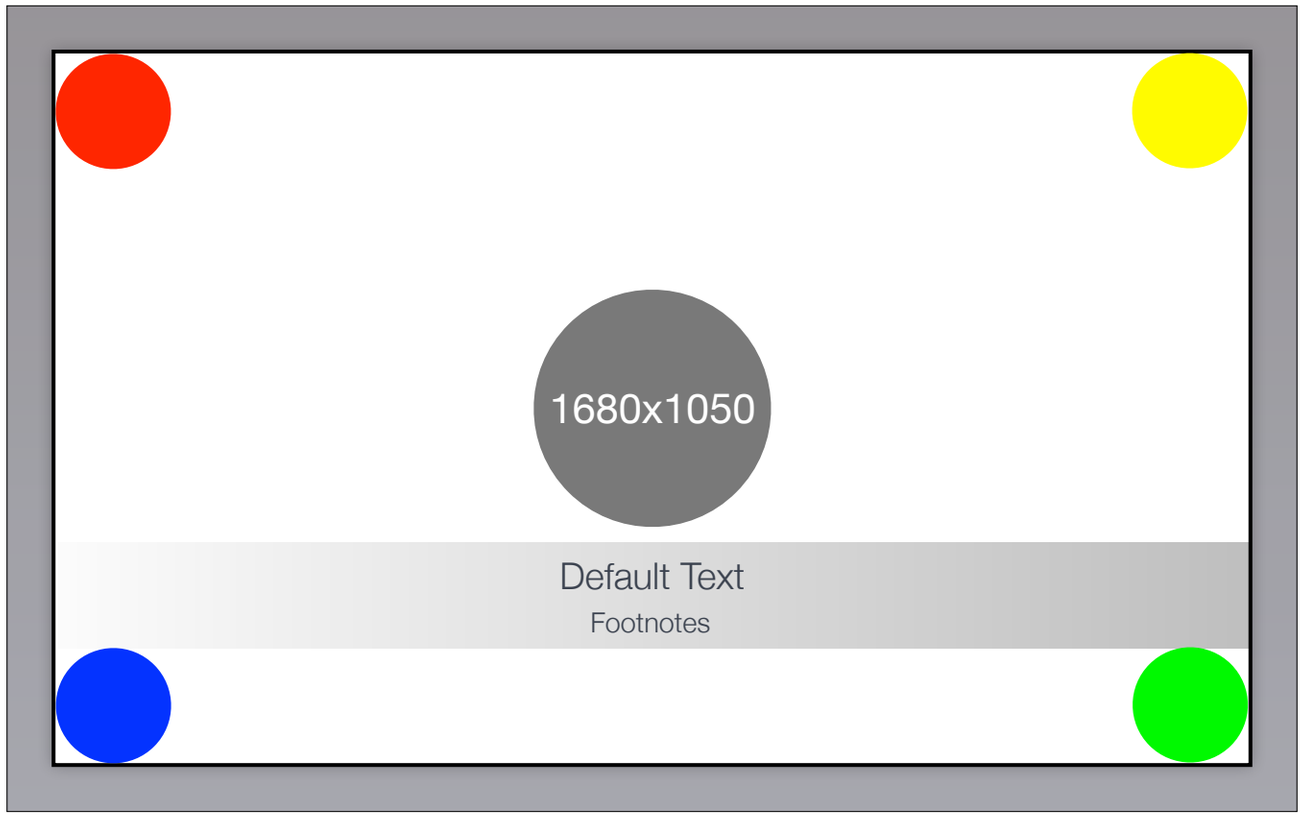
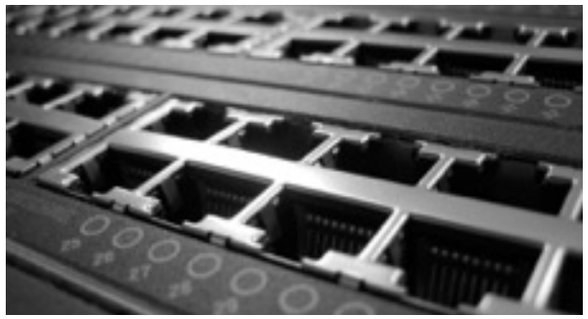

1680x1050
Default Text
Footnotes

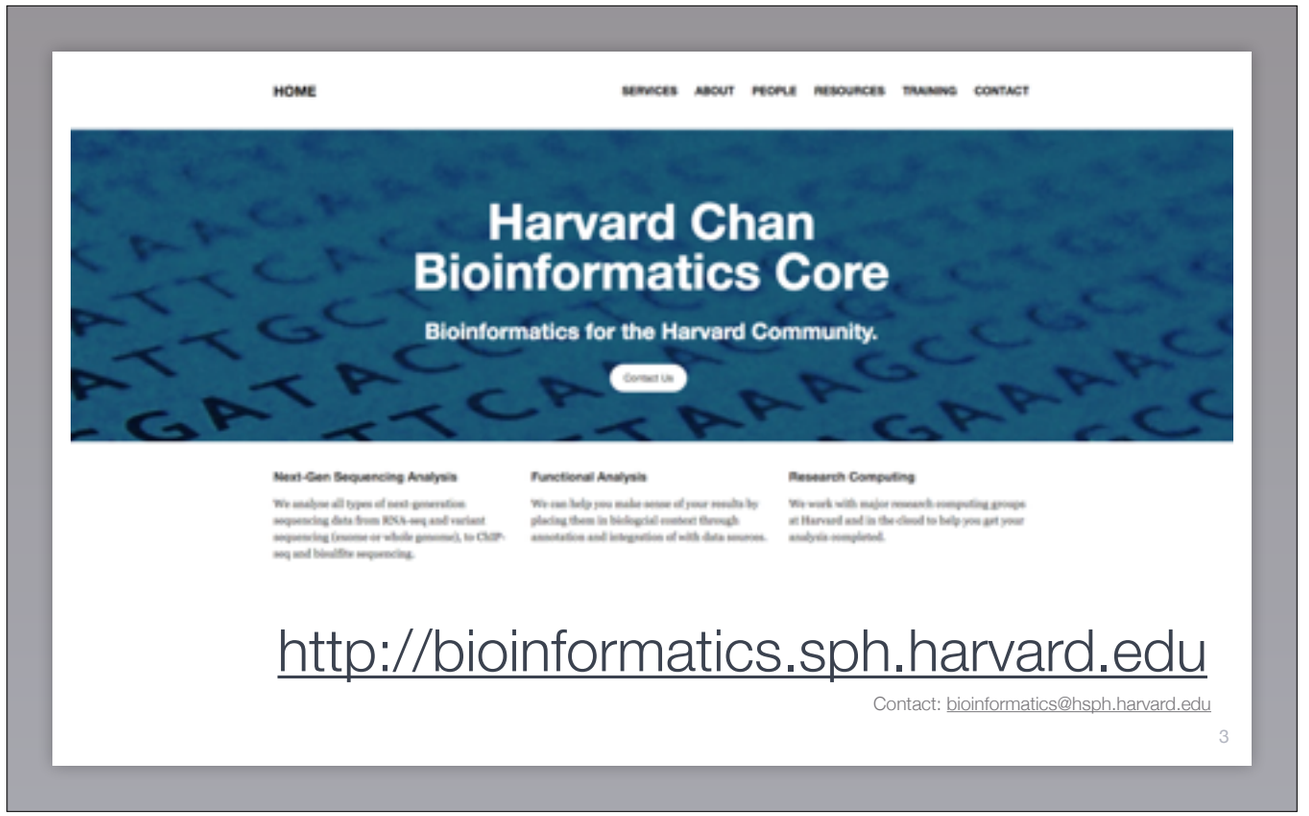


Variant Sequencing - Scaling and Metrics

Harvard Chan Bioinformatics Core (HBC)

John Hutchinson (jhutchin@hsph.harvard.edu)

Talking today about some the lessons we've learned on how to best process large amounts of sequencing data, both within a single project and across multiple projects.



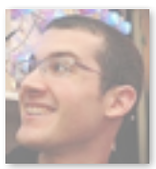
First, let me tell you a bit about ourselves. We are a bioinformatics core located at the Harvard Chan School of Public Health. You can find out more about us at our website.


Shannan Ho Sui


John Hutchinson


Oliver Hofmann


Peter Kraft


John Morrissey


Lorena Pantano


Rory Kirchner


Brad Chapman


Radhika Khetani


Mary Piper


Meeta Mistry

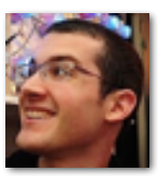
We’re a currently a team of 10 people composed of analysts, developers and trainers with Peter Kraft as our faculty advisor. John Morrissey was our research computing guru but has since moved on.


Shannan Ho Sui


John Hutchinson


Oliver Hofmann

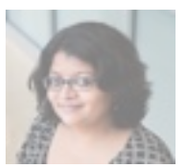

Peter Kraft

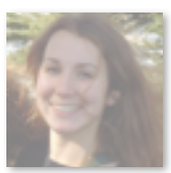

John Morrissey


Lorena Pantano


Rory Kirchner


Brad Chapman


Radhika Khetani

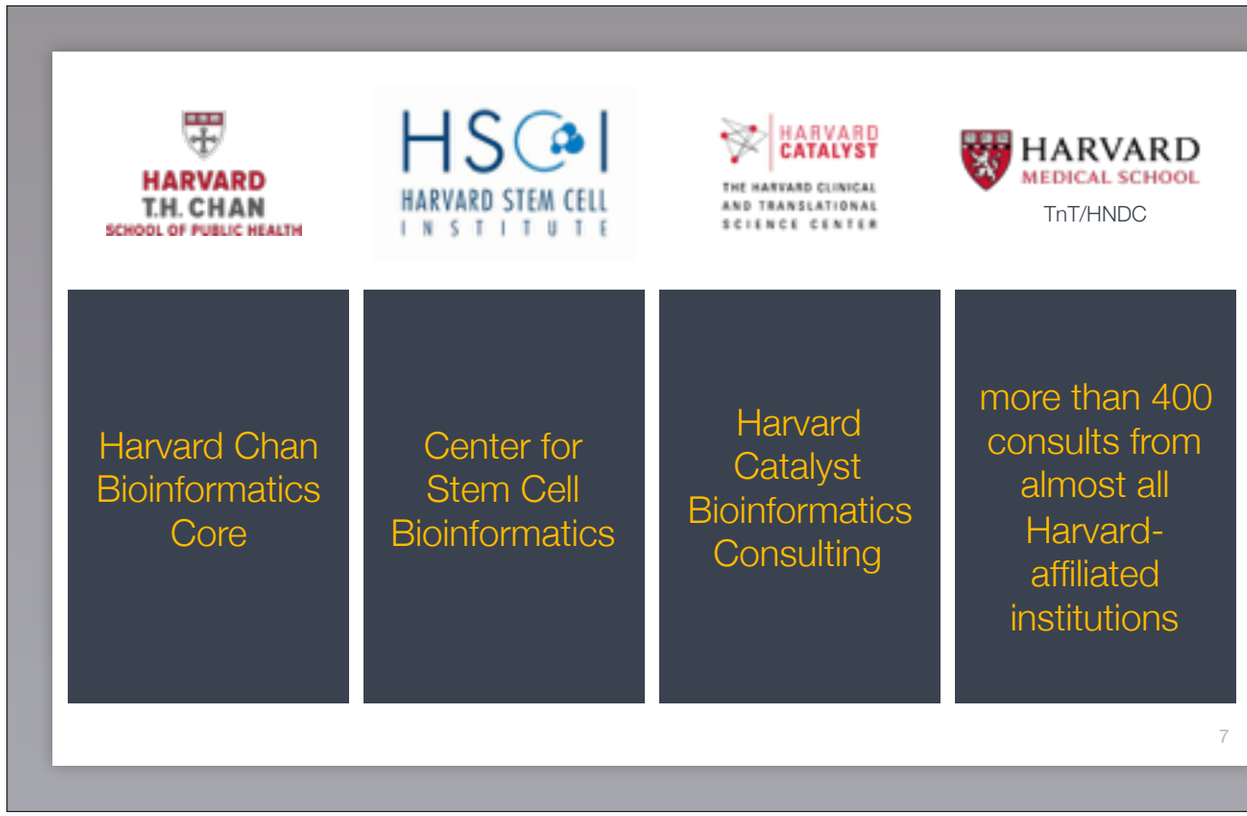

Mary Piper


Meeta Mistry

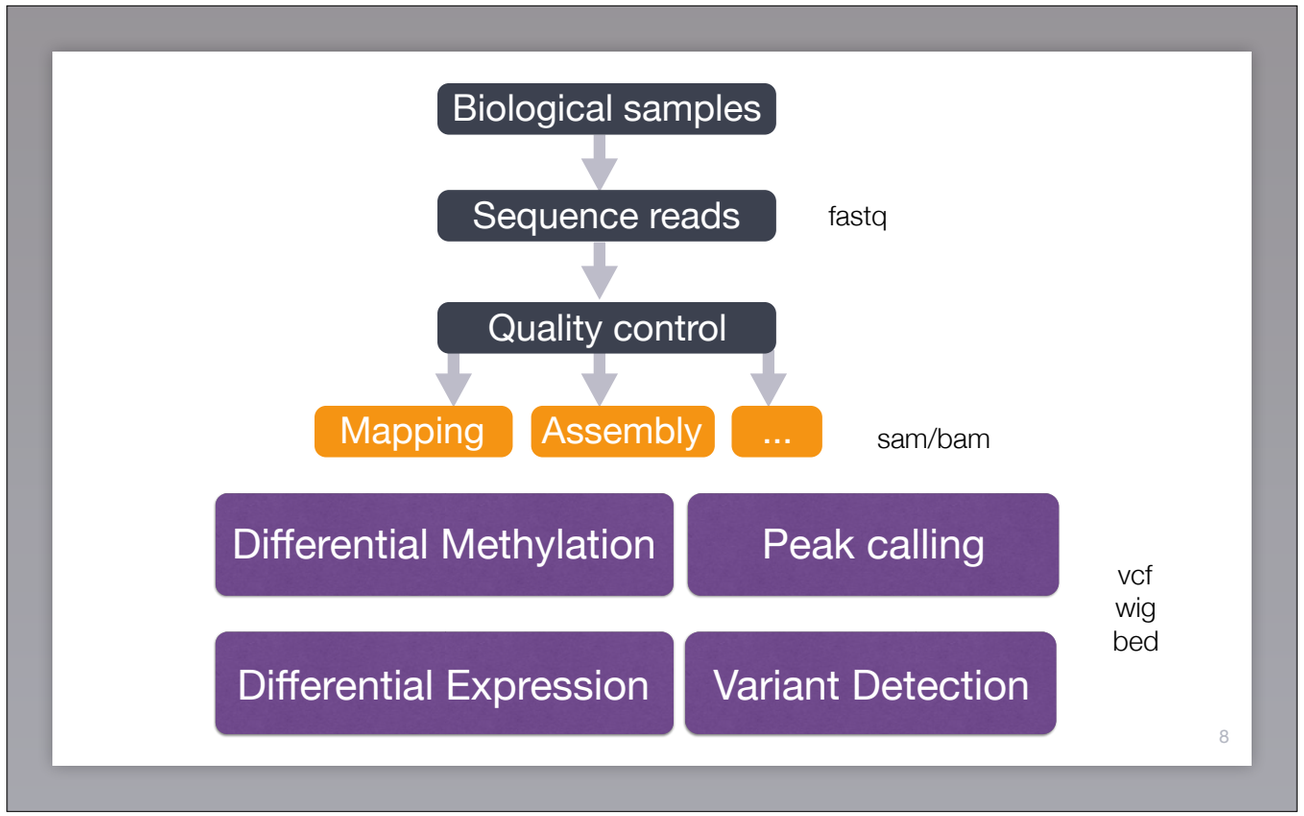
Most of what I'll be talking about here was work done by Oliver, Brad, Rory and Lorena with John helping out with the research computing.

Focus on sequencing data

Bioinformatics can be a lot of things. We **used to do everything**, from array analysis to proteomics to curation. This doesn't really scale well; re-organized to **focus on NGS data.**



We work with researchers from across the Harvard community and industry. We're a purely analytical core, with no sequencers or computing infrastructure of our own. ~**400 projects** supported in the last few years. Quickly realized we were **running into problems**, not only on the IT side but when it came to good **workflows**.



We had other issues. While most of the major file formats are stable at this point, the workflows themselves are challenging to build and maintain.

The installation hurdle

github.com/StanfordBioinformatics/HugeSeq

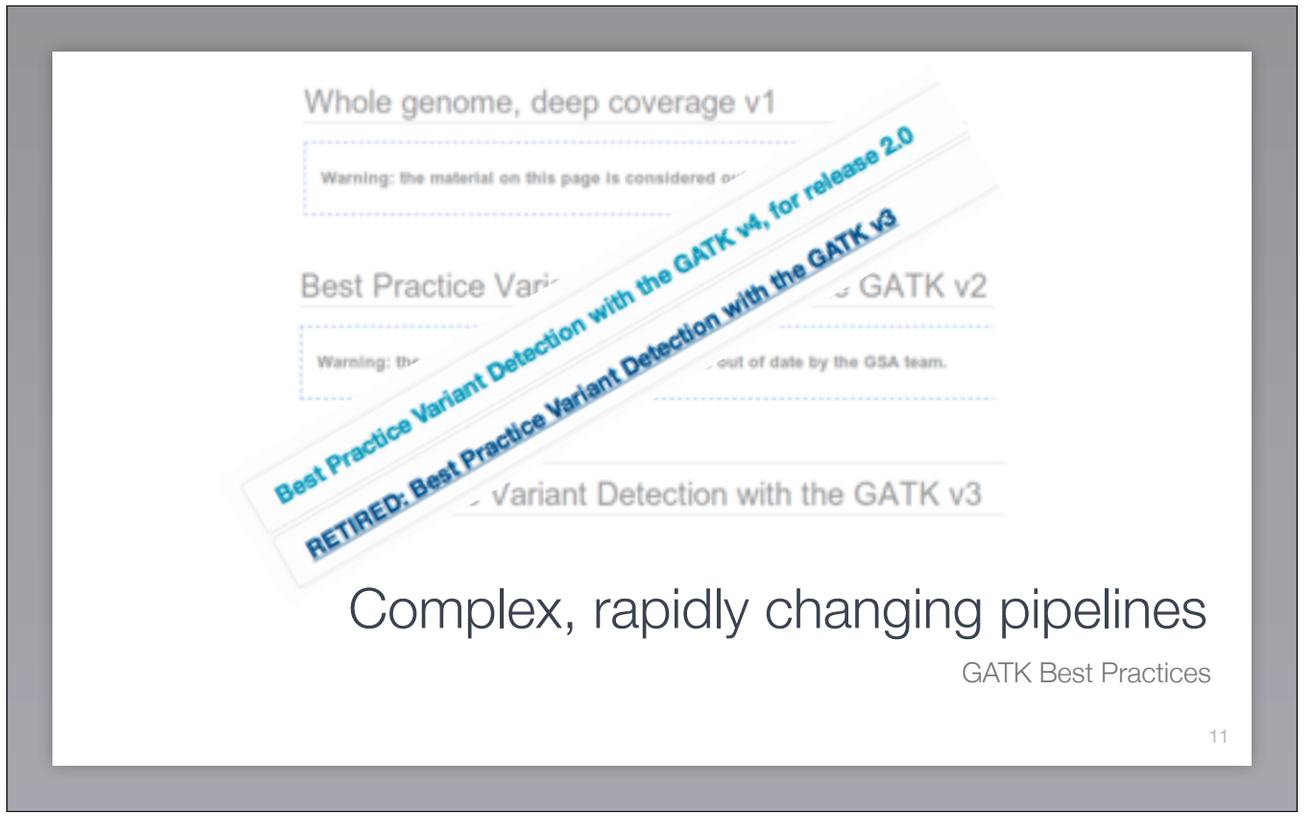
```
#####################################
# HugeSeq                           #
# The Variant Detection Pipeline    #
#####################################

-- DEPENDENCIES

+ ANNOVAR version 20110506
+ BEDtools version 2.16.2
+ BreakDancer version 1.1
+ BreakSeq Lite version 1.3
+ BWA version 0.6.1
+ CNVnator version 0.2.2
+ GATK version 1.6-9
+ JDK version 1.6.0_21
+ Modules Release 3.2.8
+ Perl
+ Picard Tools version 1.64
+ Pindel version 0.2.2
+ Plantation version 2
+ pysam version 0.6
+ Python version 2.7
+ Simple Job Manager version 1.0
+ Tabix version 0.1.5
+ VCFtools version 0.1.5
```


Lots of reasons for this. Good systems out there, but **initial installation** of frameworks can be a challenge. Dozens of external **dependencies**. And that's a problem for people with tons of programming and command line **experience**, too. We all **dread** the initial installation process and frequently **walk away from good methods** just because they are too tricky to get up and running.

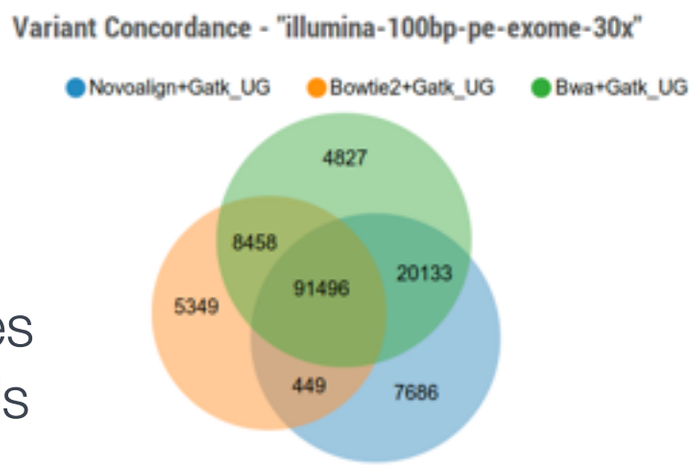
Even if we can get software installed it is often **closely tied to in-house resources** or very specific IT/cluster environments. We run analysis on different HPCs with different schedulers.



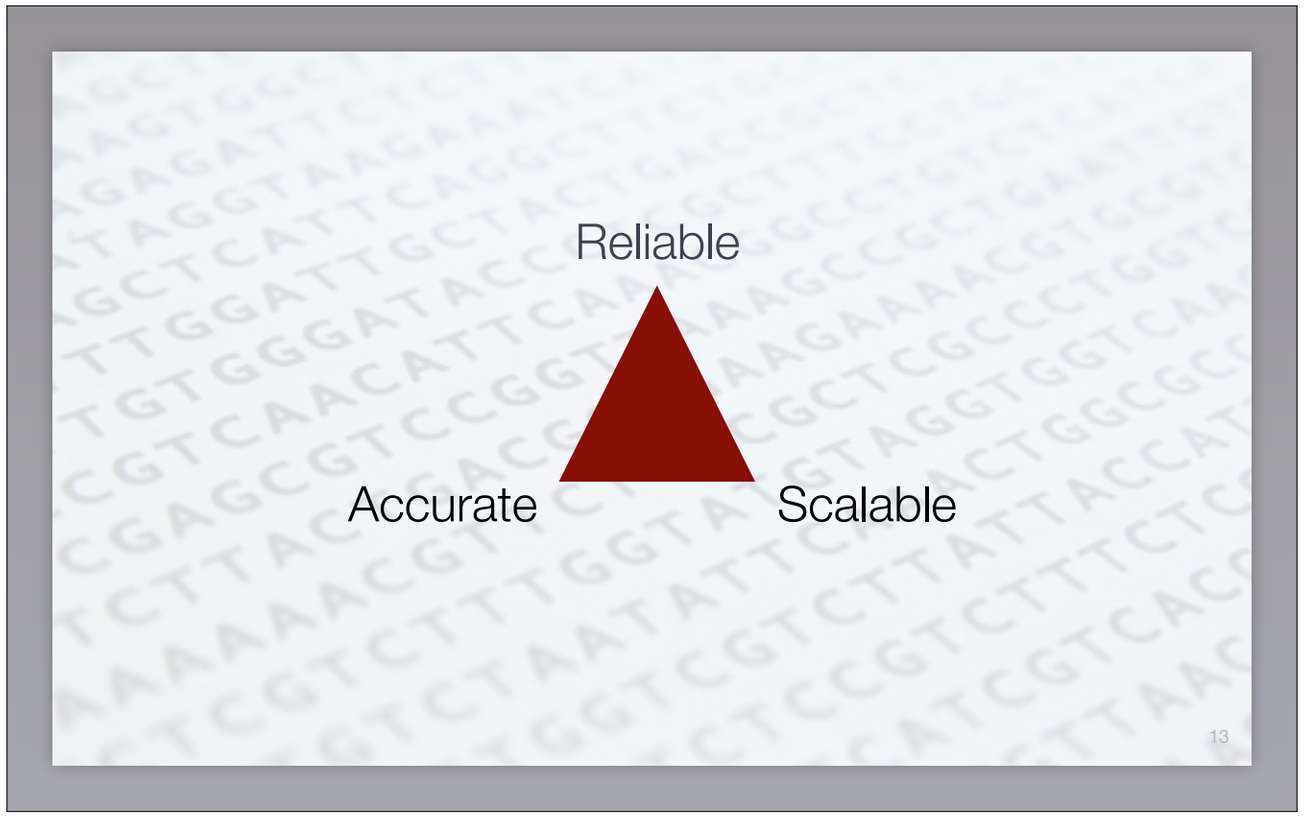
This is all made even more difficult by the way the tools are constantly being updated. It's difficult to **keep up to date** with workflow recommendations.

Quality differences between methods

www.bioplanet.com/gcat



Even if you can get everything to run smoothly there is of course the question of **what *is* the best practice** when even the choice of an aligner can result in 5% or more **discordant variant calls** -- plenty of examples at the GCAT website (Genome Comparison & Analytic Testing)



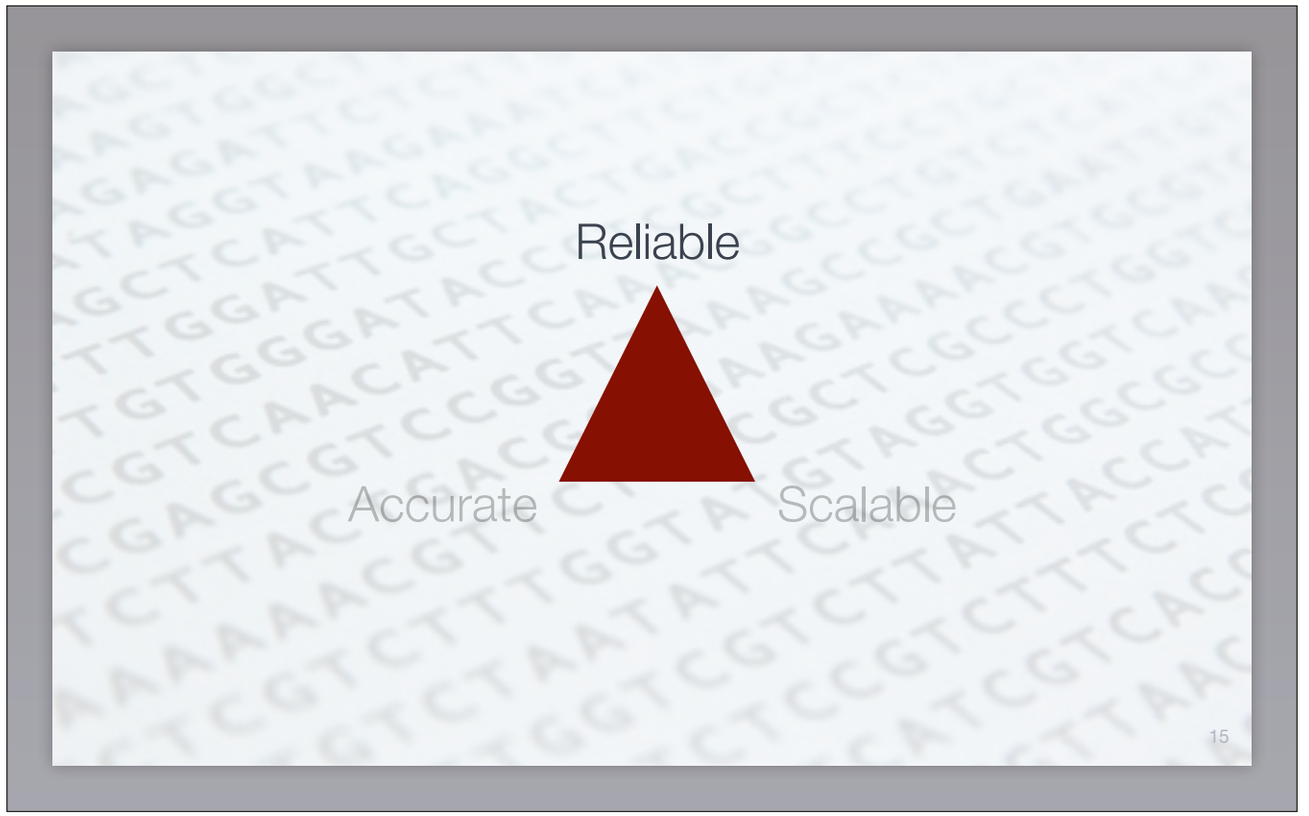
What we really **wanted** was a framework for processing NGS data that is **reliable**, **scales** well with the data, and gives us **good results**.

github.com/chapmanb/bcbio-nextgen

Our attempt at **solving some of these problems** is a framework called bcbio, where bc stands for for “Blue Collar”. bcbio is **not “the” bioinformatics NGS workflow to handle any and all data sets**.

Instead, it’s our attempt attempt to make the **most common tasks reliable and accessible**.

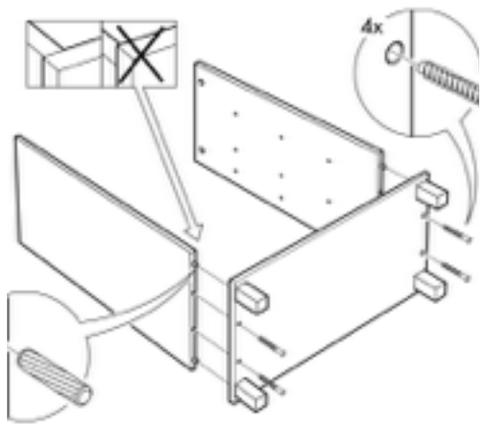
Given our core’s focus on analysis and as strong open source proponents, we decided to **work with the bioinformatics community**, **re-using methods and code bases** as much as possible.



What do I mean when I say we wanted a reliable framework? We think that good bioinformatics software should have a number of key features.

Reliable: documentation

- bcbio-nextgen.readthedocs.org



First, it would be thoroughly documented to **guide users through concepts** and **developers through implementation**. We spend ~10% of time on this, important to get right.

Reliable: “White box” software

- **Aligners**: bwa-men, novoalign, bowtie2
- **Variation**: FreeBayes, GATK, Platypus, MuTecT, scalpel, SnpEff, VEP, Gemini, Lumpy, Delly, …
- **RNA-Seq**: TopHat, STAR, Cufflinks, HTSeq, …
- **QC**: FastQC, bamtools, RNA-SeQC
- **Tools**: bedtools, bcftools, biobambam, sambamba, samblaster, samtools, vcflib, …

We package tons of methods but try to do so transparently. Brad likes to use the term **white box** — unlike the black box that you often get with commercial solutions, it’s still **wrapped** up in a (somewhat) neat package, but you can **see what’s inside** and **tinker** with it.

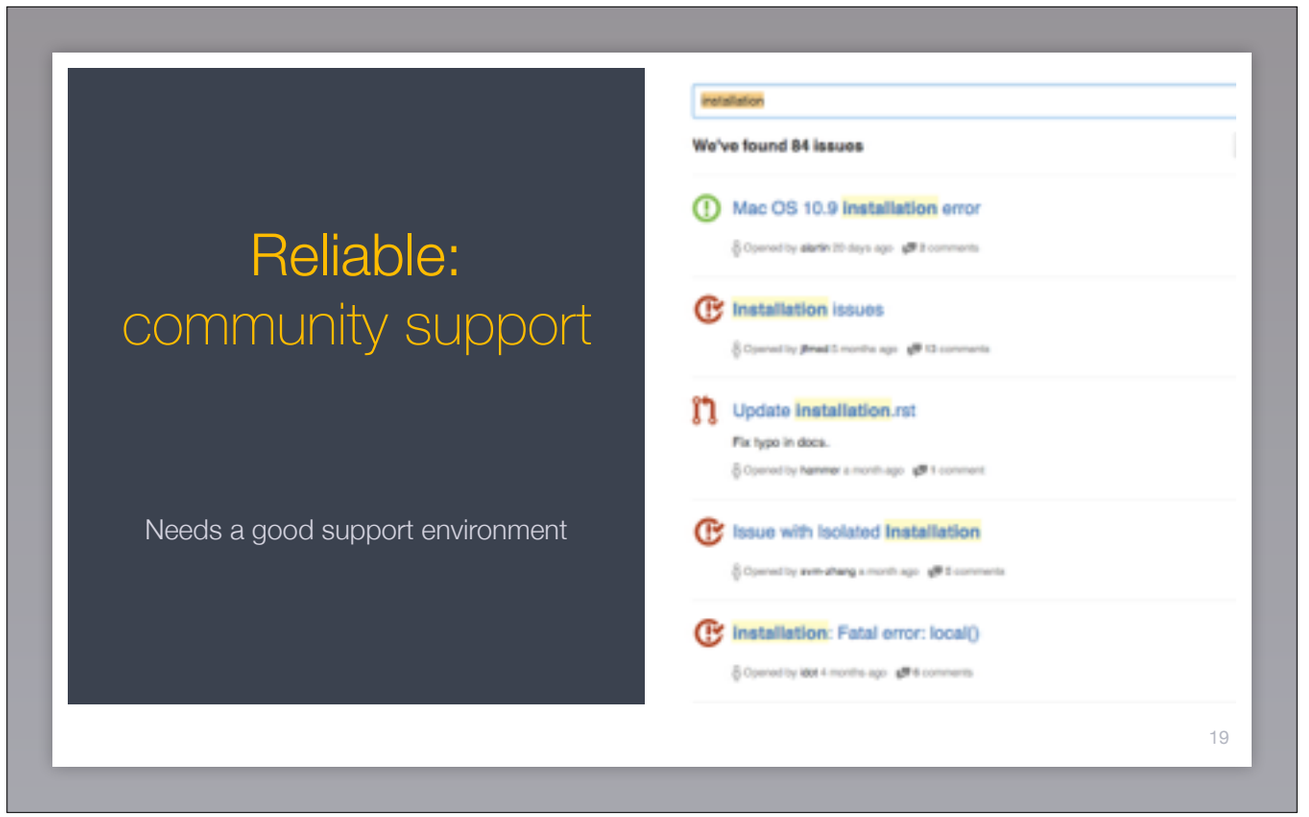
Reliable:
automated installs

Easy installation with all dependencies

Autmatic retrieval of genomes, test data

Update management

Just as important, it needs to be **accessible**, that is easy to install and use. **Bootstraps** tools, data; installs and tracks all **dependencies.** Comes with **test data** to check installation, and **push button updates.**



Of course, even if it easy to install, but **still problems in getting everything running** for the first time, or on unusual systems. Having a group of contributors helps as everyone can pitch in to answer questions.

Reliable: containers

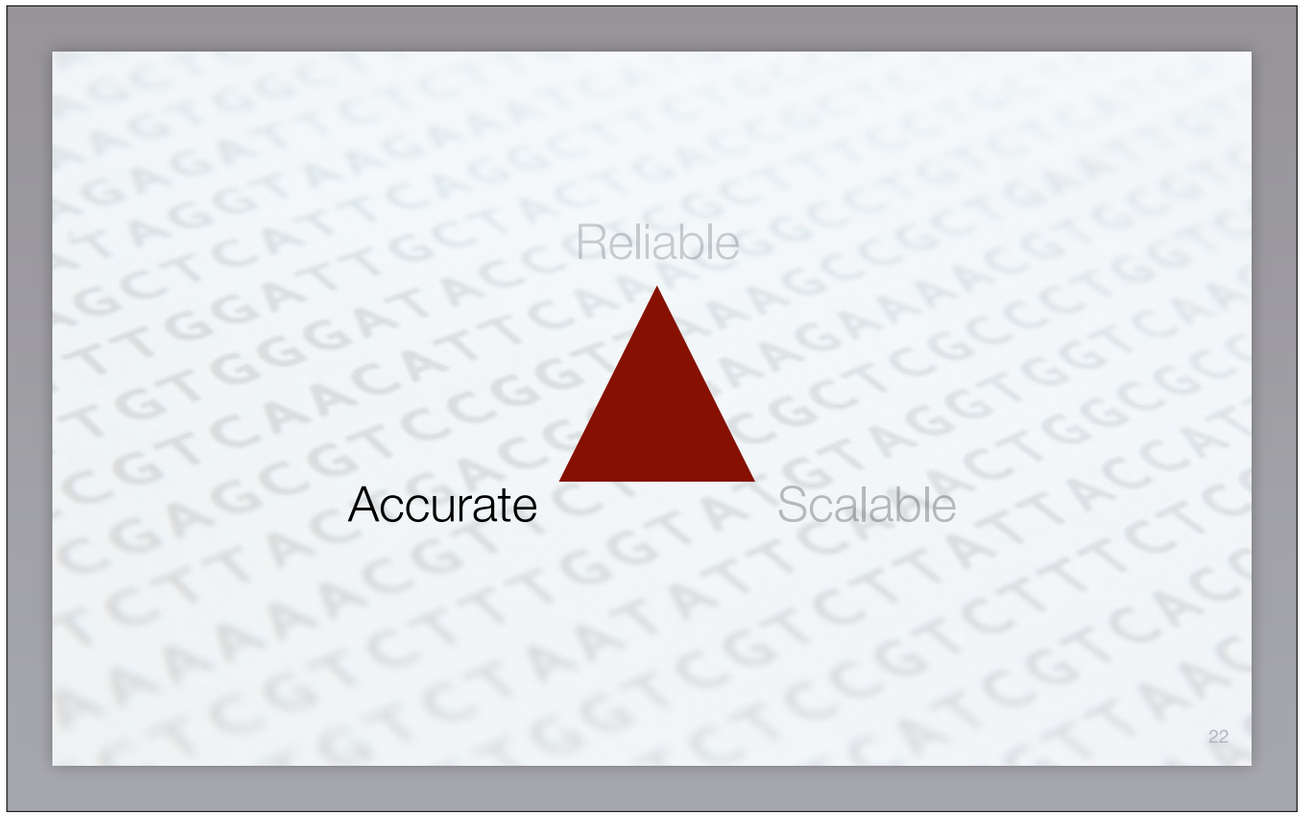
https://www.docker.io/

We further simplify the installation process by using **Linux containers** through a framework called **Docker**. These are fully isolated, single downloads which includes the data.

Reliable: provenance

- Virtual machine snapshots (Docker)
- Full logs, version tracking
- Content-Addressable Distributed File System (Arvados)

Not just for installation: we can **store snapshots** of current state at any time. Allows us to **go back months after we finished** an analysis, load the snapshot and get all our tools and configuration back to the **state** where we last stopped. We **track all configuration** files, retain complete **log files**, note **version of used external software.**



One of the most important aspect of **clinical sequencing** is being **confident about the results**. We don't need to be able to get **everything** right, but we need to have a good understanding of our **sensitivity and specificity range** if we are going to **make therapeutic decisions** based on sequencing results.

Genome in a Bottle Consortium

genomeinabottle.org

Highly confident variant calls for **NA12878**

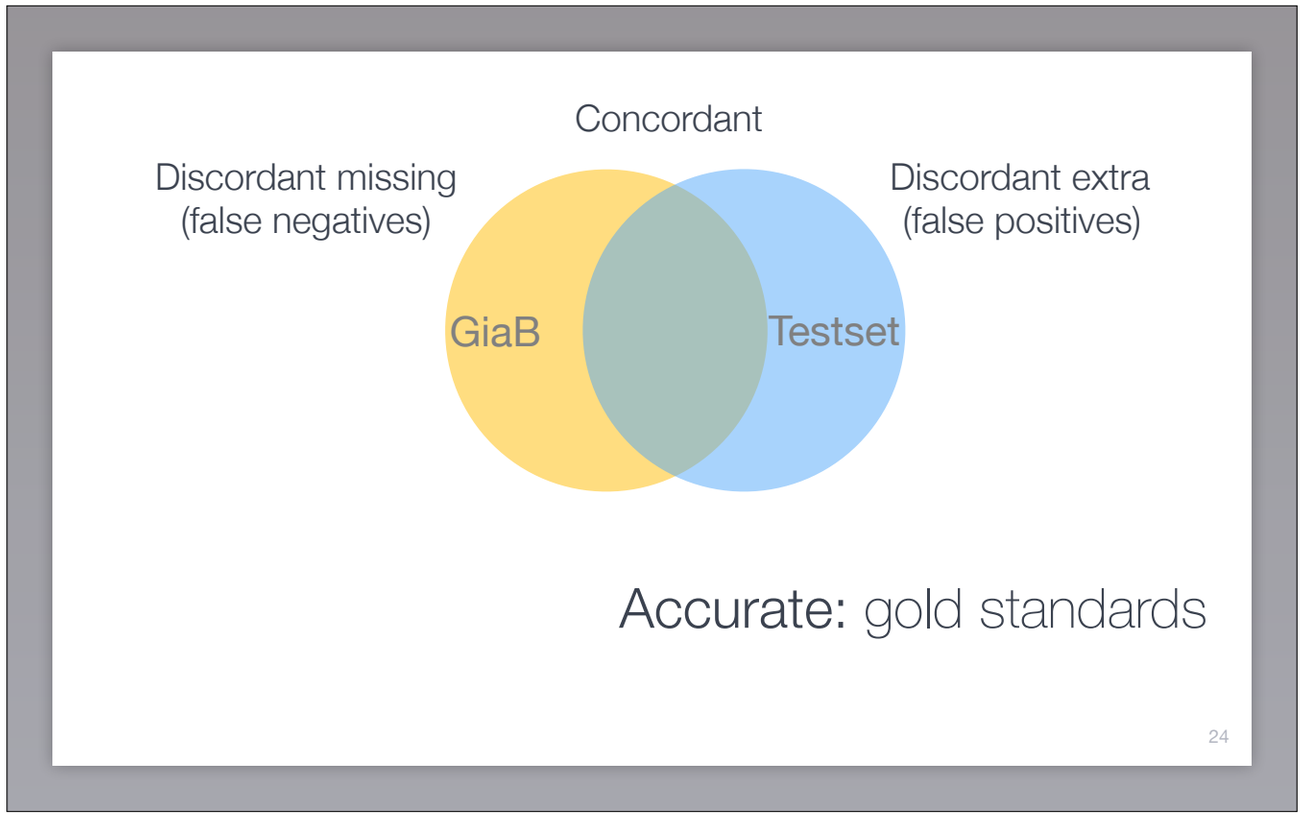
11 WGS

3 Exomes

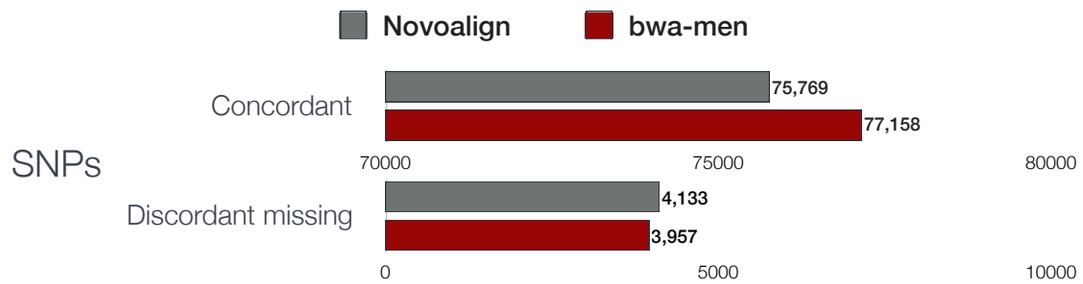
Illumina, SOLiD, 454, IonTorrent, CGI

~3 million highly confident SNPs

To have confidence in our results we need gold standards. Genome in a Bottle is a **large consortium of volunteers** around National Institute of Standards and Technology trying to create **reference materials** to improve genome analysis using DNA is sourced from **Coriell**. This is an ongoing effort to **merge and arbitrate** variant calls from multiple **technologies** and sequencing **strategies** to obtain a set of about **3 million confident variant calls**. (About **20%** of the NA12878 genome currently not callable with confidence.)



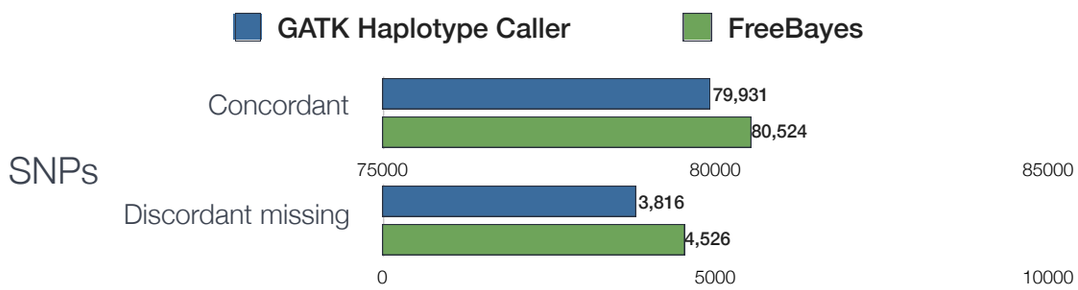
Having this gold standard lets us **optimize workflows** for better sensitivity / specificity. We can look for **false positives** (when our test set finds a variant not in the gold standard) **and false negatives** (when the test set fails to find a variant in the gold standard)**.** We utilize it to confirm workflow consistency after **code updates.** We use a similar dataset from the ERCC to test our RNAseq workflows.



Accurate: assess new tools

Quantification details: **j.mp/bcbioeval**

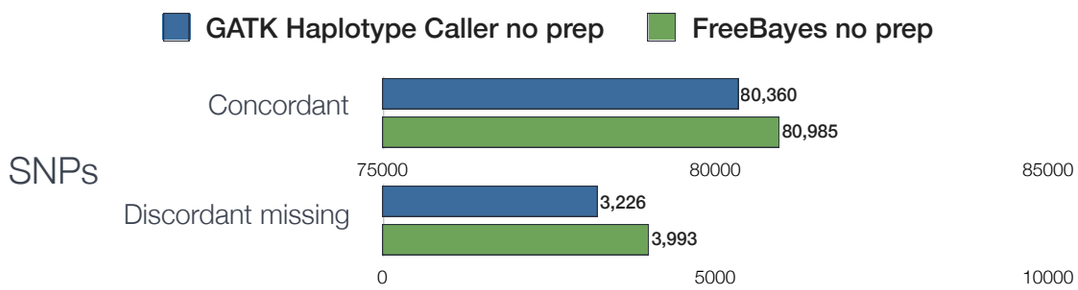
Also useful to **test new methods,** for example to compare modern **alignment algorithms**. **In our hands bwa-mem has caught up with Novoalign** for GATK 2.5 based variant calls. More concordant calls, fewer missing calls. **Key:** completely **automated**, very little overhead to re-run after changes.



Accurate: assess new tools

Quantification details: **j.mp/bcbioeval2**

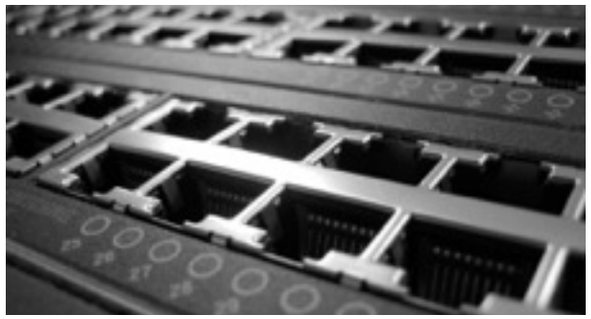
Likewise, allows us to **assess new caller** versions, often with **surprising** results. For NA12878 FreeBayes (green) **on-par** with GATK Haplotype Caller (blue).



Experiment: minimal bam preparation

Quantification details: **j.mp/bcbioeval2**

Because it's **push button** we can experiment — **modern callers already do their own read re-alignment**. Can we **skip** the standard **re-alignment** preprocessing that is part of our workflow? Turns out that FreeBayes and Haplotype Caller do not benefit from these steps. If anything we are doing better without the re-alignment, recalibration steps given good quality input data. **Significant time saving.**



Testing accuracy: Exome-seq Vendor Comparison

Now let me introduce some actual analyses. So we were fairly confident that our workflow was producing the most accurate results possible. But what about the effect of the incoming data on results?

Background

*Systematic comparison of exome-seq data from **five vendors** provided with the same **four donor DNA samples** and the following requirements:*

- *Agilent human V5 (51Mb) capture/QC (non-UTR)*
- *Illumina Library prep/QC*
- *HiSeq2000/2500 PE100*
- *Guaranteed 4.5-5.0Gb data*
- *Estimated average on-target coverage of 50x*

Four vendors used the recommended Agilent exome capture kit; the Broad used their ICE exome sequencing platform

What happens when you get the same exome-seq samples sequenced by five different vendors. What happens to the results, how do they compare?

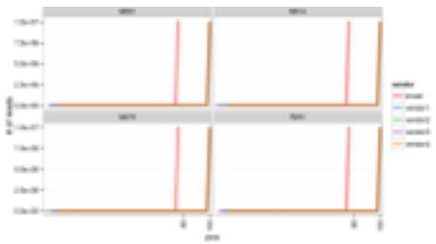
Overview

Assess the quality of capture and sequencing by performing:

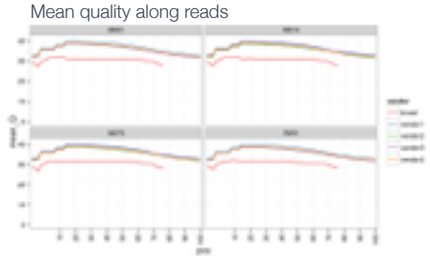
- QC of the raw reads
- Alignment and QC of the alignments
- Variant calling and filtering
- Ensemble calling to produce a consensus set of variants for benchmarking

TO assess this we analyzed the data in multiple ways, looking at the raw reads, the alignments, variant call statistics and finally the accuracy of the variant calls

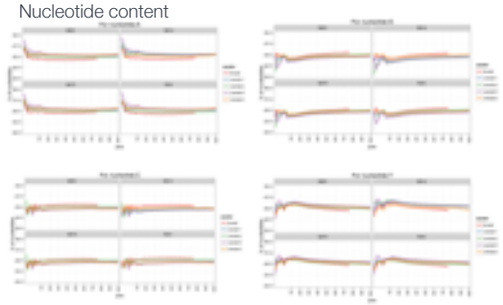
Read length distribution



Mean quality along reads

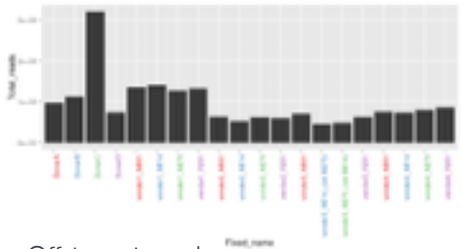


Nucleotide content

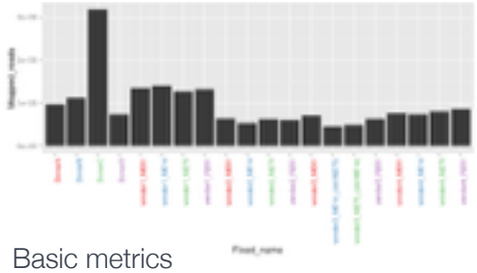


Raw sequence QC

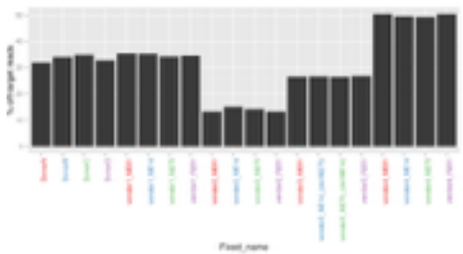
Total reads



Mapped reads



Off-target reads



Basic metrics

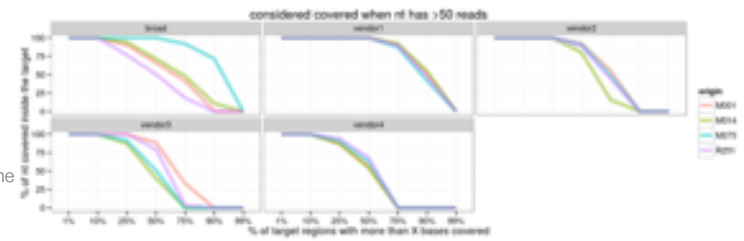
Vendor	Total reads	Mapping rate	Duplication rate	Median insert size
Broad	152,092,495	0.9888	0.2392	168
Vendor1	134,867,047	0.9930	0.0812	194
Vendor2	60,488,200	0.9979	0.0804	206
Vendor3	57,855,058	0.9964	0.0529	216
Vendor4	79,486,159	0.9977	0.0448	161

Alignment QC

Looked at a number of basic read mapping statistics such as the number of total reads, mapped reads, off-target reads and the mapping and duplication rates. All were similar, though one vendor large variations in total reads and one vendor had a lower number of off-target reads.

Coverage distribution of target regions

- Determined percentage of nucleotides in each target region with at least least 50X coverage **(y-axis).**
- The quantiles of the distribution of these percentages for all target regions are plotted on the **(x-axis)**
- So, a point at 10% on the x-axis and 60% on the y-axis means that 10% of target regions has <= 60% of nucleotides with 50X coverage



Alignment QC

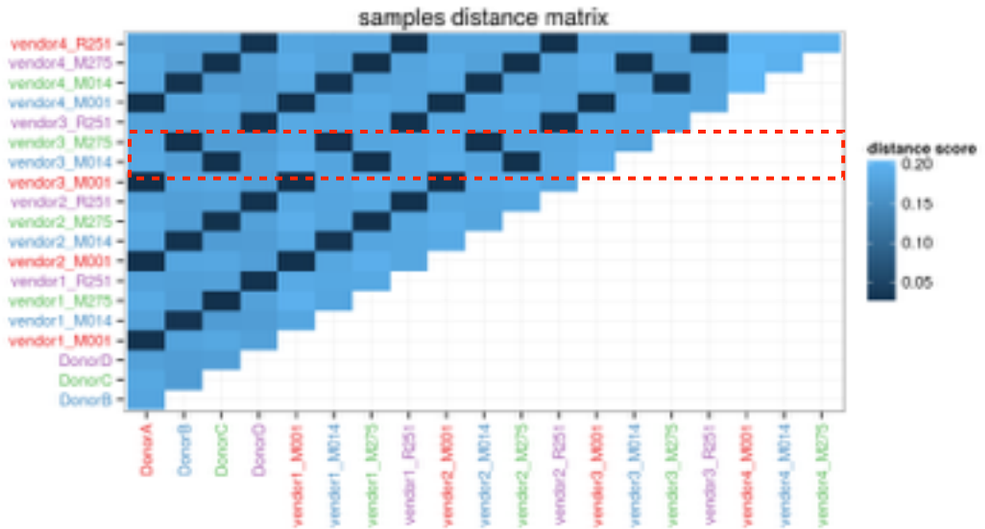
We also looked at more sophisticated metrics, including the coverage distributions of the exome pulldown target regions. Here, vendor 1 and 2 performed best.

For vendor 1, 75% of the target regions had almost 100%^ of their nucleotides with 50X coverage. 90% of vendor 1’s target regions had 50% of their nucleotides with 50X coverage.

Variant calling

- Using bcbio
- Used Agilent exome target bed file to analyze all variants inside those region
- Variants called using the GATK HaplotypeCaller using GATK best practices with hard filtering

We used bcbio with GATK best practices to call variants in these samples within the exome target regions.



- qSignature: sample similarity using VCF files
  - http://sourceforge.net/p/adamajava/wiki/qSignature%20SignatureCompareRelatedSimple/
  - **detected two samples with incorrect labels for vendor3**

Oen of the first things we did was to use the resulting variant call files to check for sample swaps using qSignature. We picked up two samples with incorrect labels from one vendor. While we had it easy and were able to compare to identical samples, this method could theoretically also be used to verify related samples such as family trios.

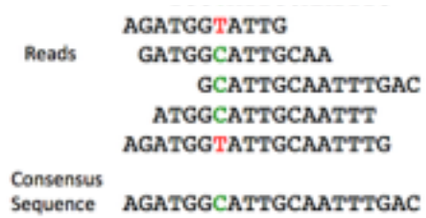
Vendor	Total variants	% in dbSNP	Heterozygous	Homozygous	Ratio(het/ hom)
Broad	42532	98.30	26,215	16,317	1.6067
Vendor1	45600	97.82	28,696	16,904	1.6978
Vendor2	44794	98.13	27,918	16,877	1.6543
Vendor3	44993	98.00	27,990	17,003	1.6463
Vendor4	44590	98.03	27,636	16,954	1.6301

Variant metrics per vendor

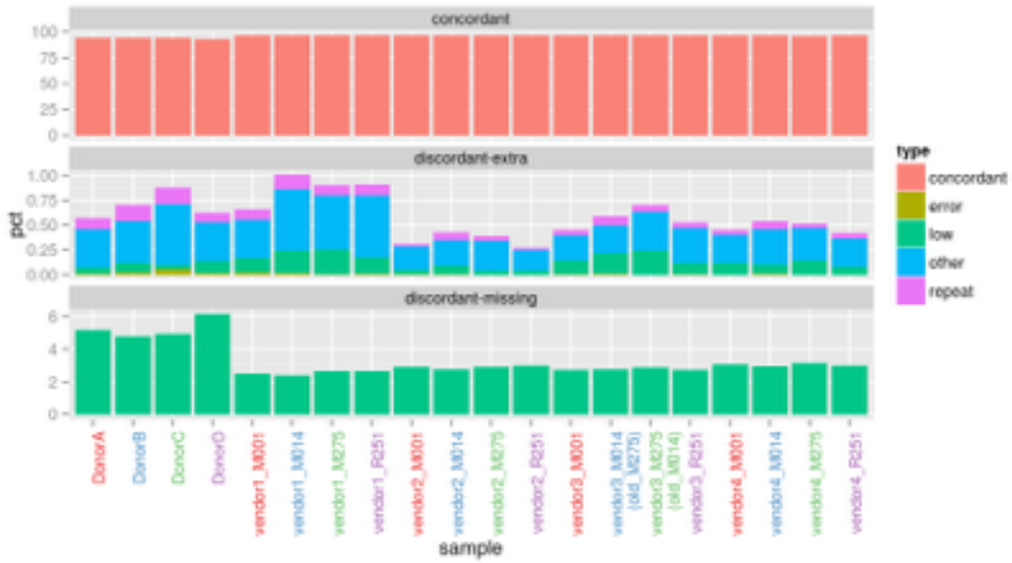
Averaged across the four donor samples

Validation against a “truth” set

- Performed Ensemble calling across vendors to produce a consensus set of variants
- Considered “real” variants as any position that had a variant in 3/5 vendor samples



Of most importance, are the SNP calls accurate? This is a challenge as we had no real benchmark or “truth” set. Instead, we did the next best thing and used consensus calls from majority of vendor samples to call variants.



Variant comparison

Using common Agilent & Broad regions (41 MB)

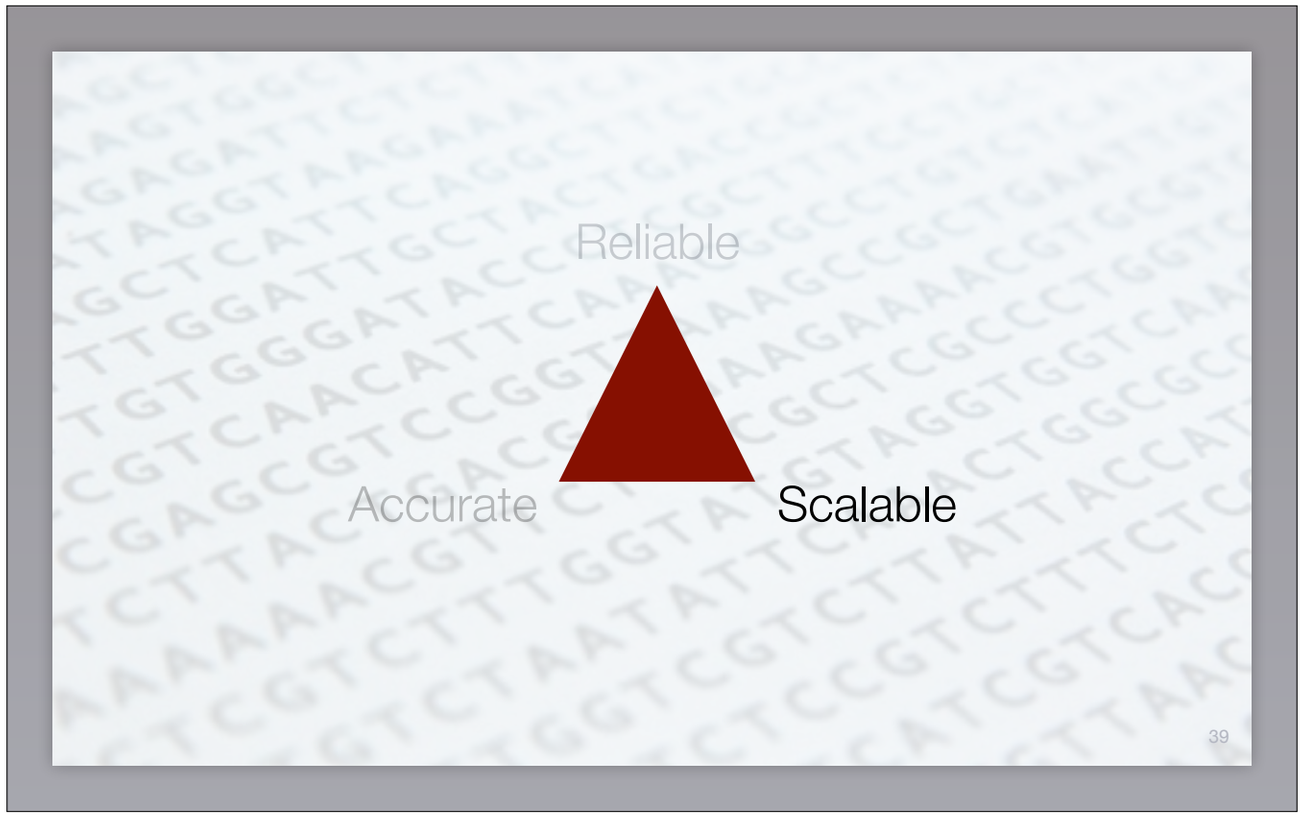
We then compared this truth set back to each individual sample

- Concordant: same genotype as the truth set
- Discordant-extra: calls made by the vendor that are not in the “truth” set (false positives)
- Discordant-missing: variants in the “truth” set that are missed by the vendor (false negatives)

When possible we assigned a likely reason to each of these discordant calls to see if there were any patterns.

- Error: error-prone regions as defined by Content-Specific Sequencing Errors (https://code.google.com/p/discovering-cse/)
- Low: low coverage variants
- Repeat: variant inside repeat region
- Other: None of the above

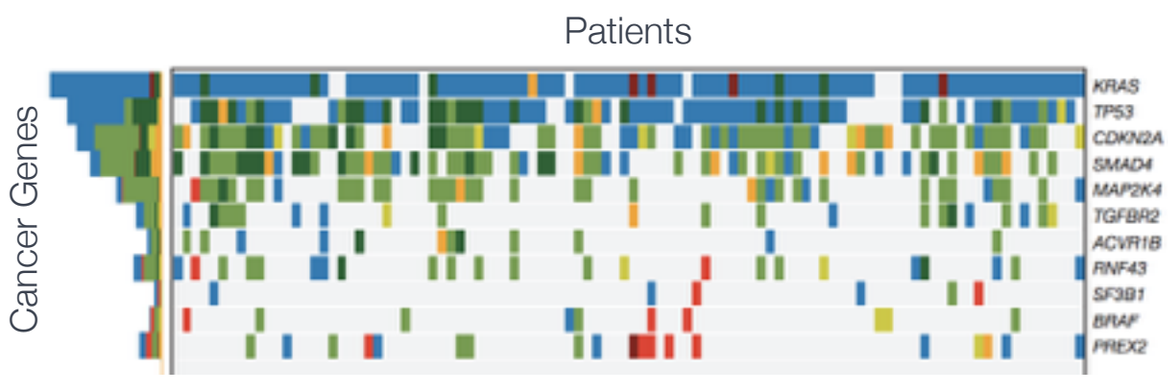
In the end, most of the vendors performed well, though two showed higher and more consistent coverage. But studies like this are we advise clients to run test samples through a facility before committing to any irreplaceable samples.


Reliable
Accurate
Scalable

Scalable: building the right architecture

LEt's switch back to bcbio again and talk about the final corner of scalability. That is, we want a framework that is both **robust** and **fast** enough to handle current NGS data sets.

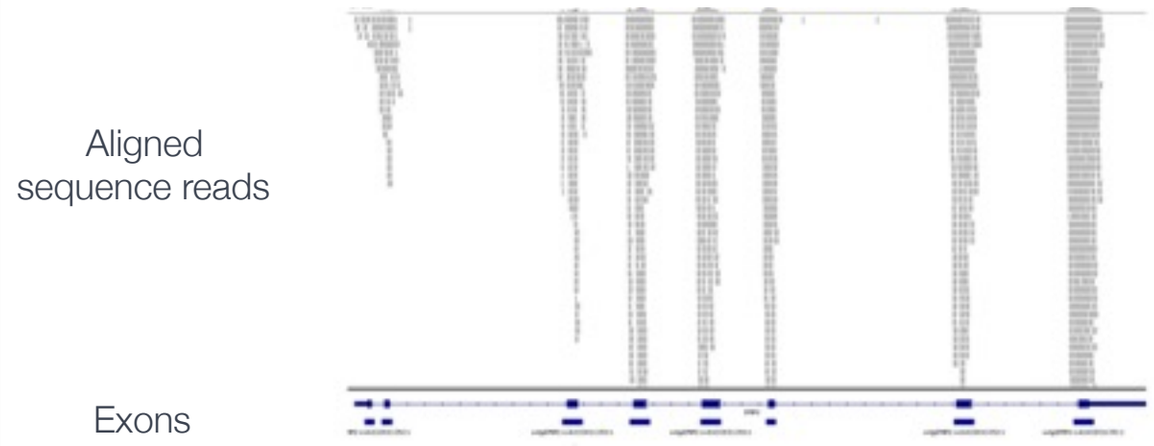
Speed is critical not only because people want their results yesterday, but because only if the system is reasonably fast can we actually iterate to e**xplore multiple approaches to an analysis.**



Gene panel sequencing and cancer

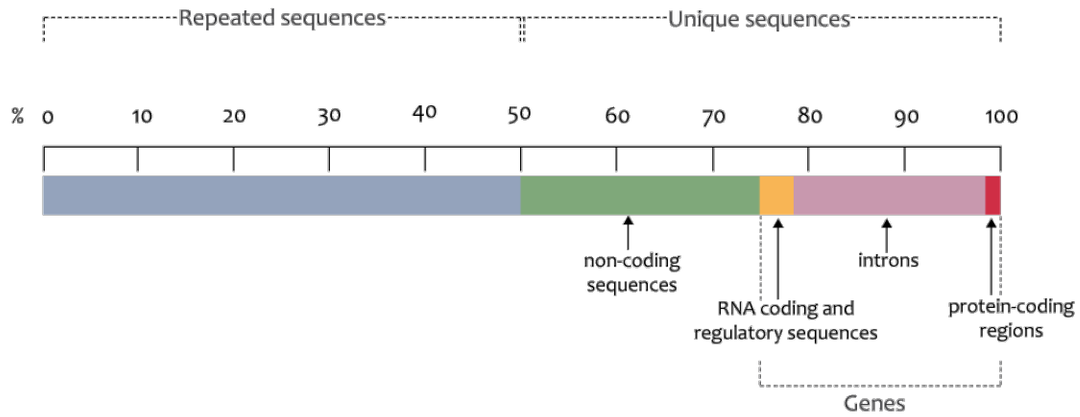
Waddell et al, Whole genomes redefine the mutational landscape of pancreatic cancer, Nature 2015

bcbio supports **RNA-Seq and all kinds of re-sequencing studies** and has little problem handling gene panels for thousands of patients.



Exome sequencing

We have also used it for a number of exome-seq studies including work with Peter **Kraft** on predisposition to breast cancer in **6000 participants** of the Nurses' Health Study. WE expanded on that work with bcbio



Whole genome sequencing

But **increasingly** we are being asked to support **whole genome analysis.**
The only problem with that: it's **quite a bit more data**. Exome-seq encompasses around 2% of the genome.

Our challenge: 1500 WGS samples

Let me take you through our experience scaling up to tackle a whole genome project for the first time. This was work with Rudy Tanzi at Mass General to study early onset of Alzheimer's disease via Whole Genome Sequencing.

Requirements per sample

100 GB storage

300 GB during processing

~1300 Core Hours

Back of the envelope calculation of the number of Core hours and storage needed per sample. This varies a bit depending on the algorithms used, but it's a good **ballpark figure**.

Requirements for 1500 WGS

~150 TB of storage

~300 TB of active storage

~2 million core hours

(200 single core years)

(5 months on 512 cores)

Scaling that up to 1500 samples amounts to a fair amount of compute and storage. This doesn't run cheap, particularly if you are running this on a shared/public cluster environment.

Engineering challenge: nothing works at scale

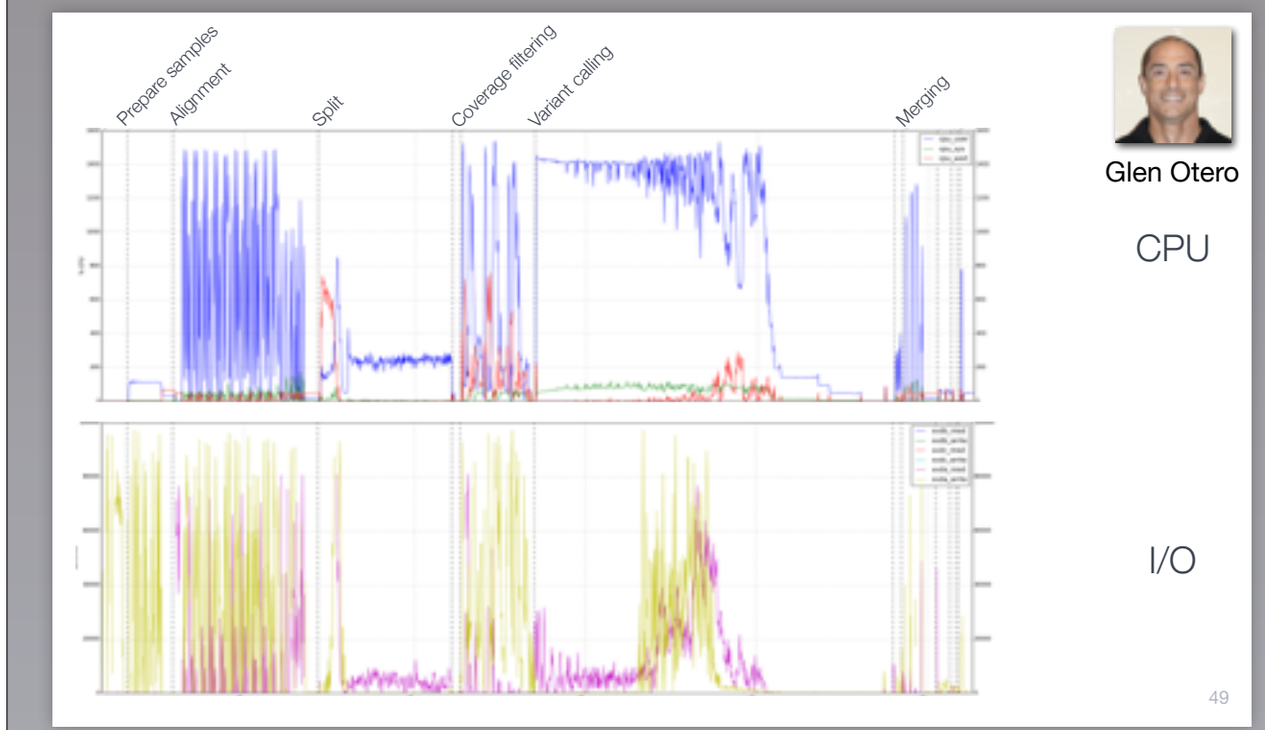
We took this project **knowing** that we'd **have to make changes** to the workflow, but the reality is that — out of the box — **nothing works** at this scale. Nothing. Schedulers failed, we overloaded the **network** on the local cluster. Killed **file servers**. Got **memory errors** on algorithms. Our initial WGS samples weeks to finish. This is when we started reaching out to colleagues.

512 Core Cluster
3GB RAM/Core
512TB Lustre Storage
Infiniband Network

Scalable: infrastructure

http://www.dellhpcsolutions.com/dellhpcsolutions/static/genomics.html

Luckily Dell was in the process of designing a cluster for NGS analysis and had an offer: we help them with testing their new NGS system and run benchmarks, we got to use spare cycles.

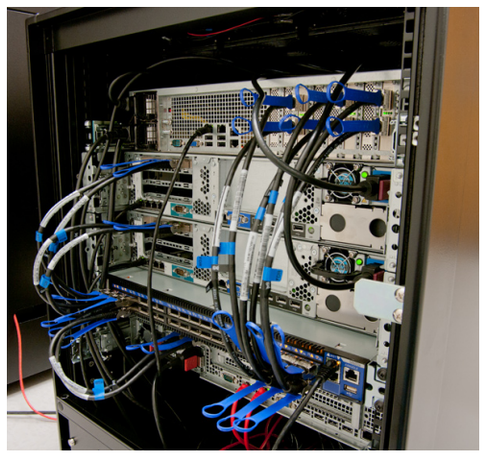


Perhaps more importantly, this gave us **access** to another ‘community’ of dedicated **research computing specialists** who were able to measure just about everything in detail without ‘noise’ from other jobs on a cluster.

This enabled us to **benchmark bcbio** much more precisely allowing us to **check** CPU, memory, IO **requirements** for the different processing steps and then redistribute load where needed and optimise the cluster configuration for each step to avoid bottlenecks.

Scalable: more bandwidth

- From 1G Ethernet to Infiniband



To speed things up, we used Infiniband connects to communicate with the filesystem.

	Bottleneck	1 WGS 16 Cores Lustre	1 WGS 96 Cores Lustre	1 WGS 96 Cores NFS	30 WGS 480 Cores **Lustre**	30 WGS 480 Cores **NFS**
alignment	CPU/Mem	4.3h	4.3h	3.9h	4.5h	6.1h
post-process	IO	3.7h	1.0h	0.9h	7.0h	20.7h
variant calling	CPU/mem	2.9h	0.5h	0.5h	3.0h	1.8h
post-process	IO	1.0h	1.0h	0.6h	4.0h	1.5h
total		11.9h	6.8h	5.9h	18.5h	30.1h

Scalable: parallel filesystems

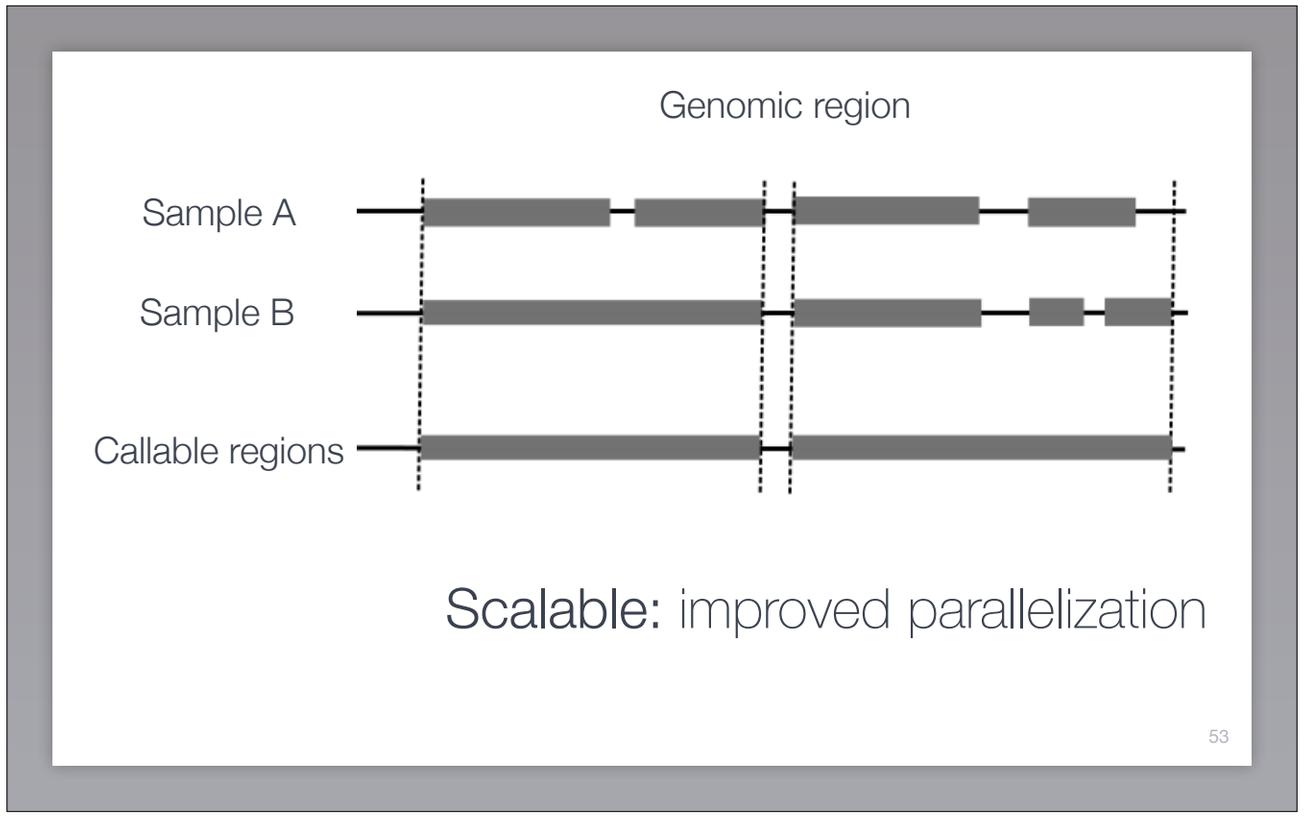
j.mp/bcbioscale

We also optimized our filesystems. At 30 WGS samples and 480 cores **NFS** starts becoming limiting for post-processing. A Lustre based parallel filesystem held up better.

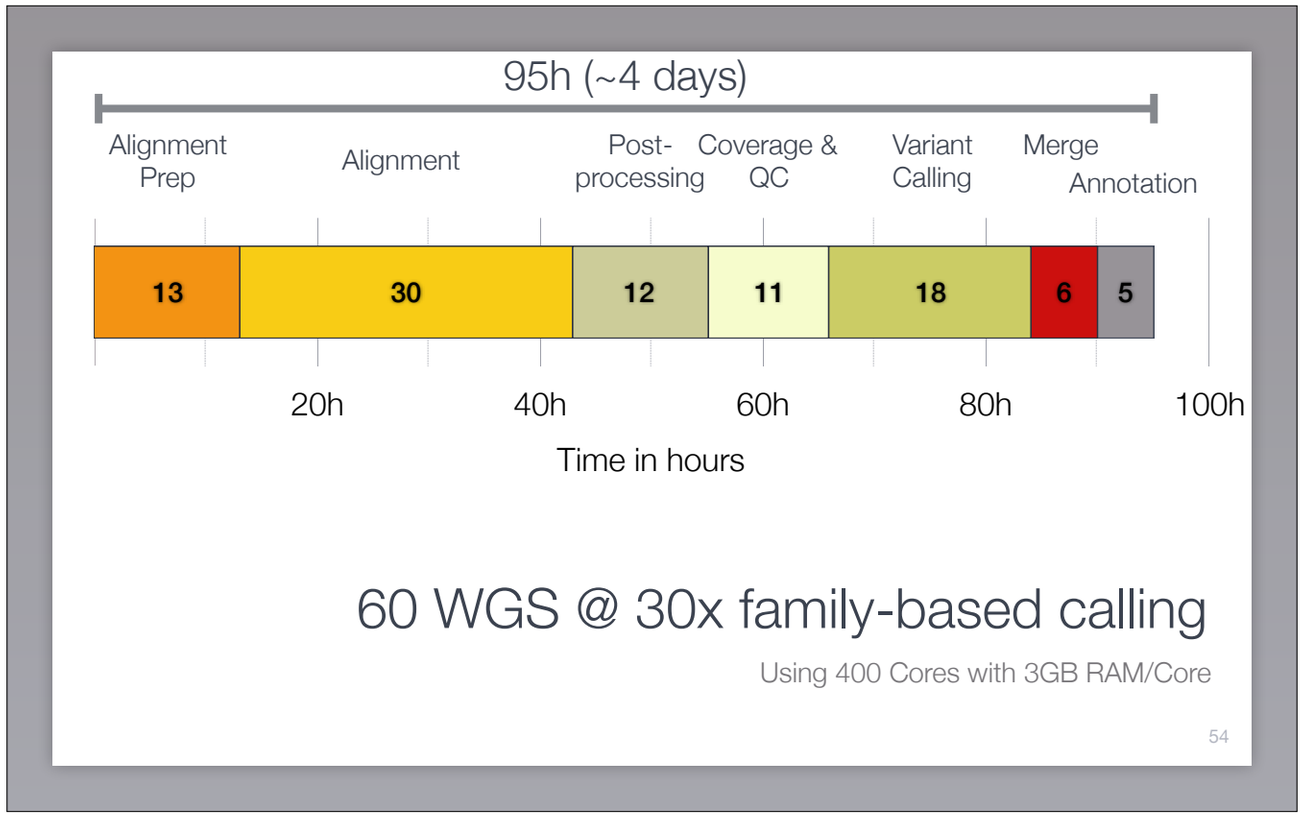
```
("{bwa} mem -M -t {num_cores} -R '{rg_info}' -v 1 "
 "  {ref_file} {fastq_file} {pair_file} "
 "| {samblaster} "
 "| {samtools} view -S -u /dev/stdin "
 "| {sambamba} sort -t {cores} -m {mem} --tmpdir {tmpdir}"
 "   -o {tx_out_file} /dev/stdin")
```

Scalable: avoid intermediates

And even with a parallel filesystem you get **slowdowns** so we changed the workflow to **avoid IO access** as much as possible, **streaming** all data where tools allow it and no intermediate result needs to be stored.



We did other non-hardware based tweaks. For example, instead of splitting out subtasks up by a small number of **chromosomes** we split them out by the thousands of separate **genomic regions** that had **coverage** in all samples.



Total of **4 days**, or about **2h/sample** using 400 Cores. Or about **100 WGS** (or 25 cancer genomes)/**week** with a best practice workflow.

This is a big step forward but there is a lot of room for improvement. These samples were run on a 512 core optimized server that runs north of **$800,000 USD**. And that's before salaries, and power and cooling for the server.

Thank you for listening!

jhutchin@hsph.harvard.edu

@ordinator

github.