Engineering Firehose: The First Year

Daniel DiCara, Gordon Saksena, Raktim Sinha, Rui Jing, Douglas Voet, Michael Noble

Broad Institute of MIT and Harvard, Cambridge

Http://GDAC.BroadInstitute.org

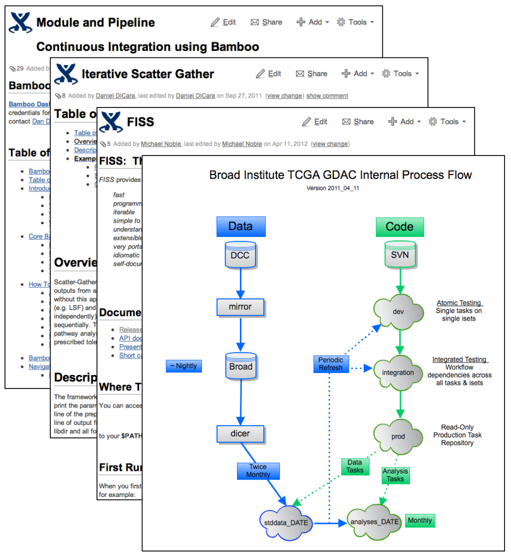
Abstract

The Broad Firehose pipeline was born of the desire to systematize analyses from The Cancer Genome Atlas (TCGA) pilot phase and scale their execution to the dozens of remaining diseases to be studied. To realize this goal, while retaining flexibility to incorporate emerging algorithmic advances, it was necessary to rapidly evolve research codes, data processing scripts, and infrastructure into mature components that reliably process terabytes of data on a monthly basis; all towards the additional aims of high transparency and scientific credibility.

Here we detail how that process unfolded in the first year of Firehose operation. It has led to our view of Firehose as a virtual data factory, subject to the same production constraints of timeliness and reliability under which physical factories function. We review the substantial progress made in several key areas, including semantic clarity, robustness, transparency, pinpoint control and easy extensibility through scripting, and scalability through automation and parallelism.

We discuss how these improvements have not only allowed us to fulfill the original aim of providing a monthly package of standard analyses results, but also to create additional value for the TCGA. Three examples are our twice-monthly standardized data packages, opportunistic runs for analysis working groups, and our positive influence on other key processes such as mutation data flow.

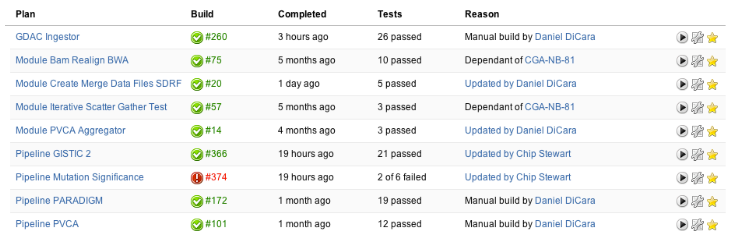
Clarity & Transparency: Tools and Process Docs


Module and Pipeline
Continuous Integration using Bamboo
Edit
Share
Add
Tools
Iterative Scatter Gather
Added by Daniel DiCara, last edited by Daniel DiCara on Sep 27, 2011 (view change) show comment
FISS
5 Added by Michael Noble, last edited by Michael Noble on Apr 11, 2012 (view change)
FISS provides
fast
iterable
idiomatic
Broad Institute TCGA GDAC Internal Process Flow
Version 2011_04_11
Data
Code
DCC
SVN
mirror
dev
Atomic Testing
Single tasks on single isets
~ Nightly
Broad
Periodic Refresh
integration
Integrated Testing
Workflow dependencies across all tasks & isets
dicer
prod
Read-Only Production Task Repository
Twice Monthly
Data Tasks
Analysis Tasks
stddata_DATE
analyses_DATE
Monthly

Clarity & Transparency

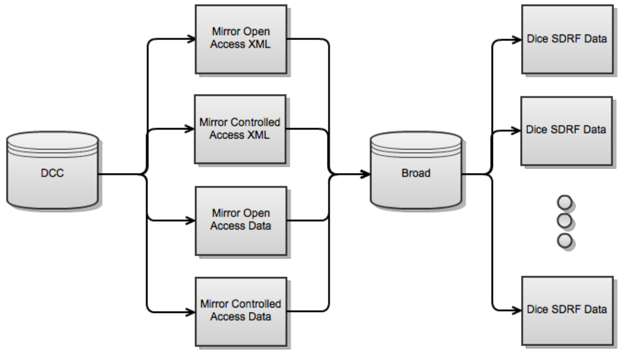
In terms of clarity, a number of code enhancements have been made to abstract business logic out of procedural code spread across multiple files into concise classes and methods. This has facilitated greater code reuse, exposed bugs, and increased transparency. Furthermore, hard-coded features pervasive throughout some of our code have been placed in configuration files. This permits the overriding of default behavior in a non-invasive fashion for testing purposes or otherwise. Finally, and most importantly, we have significantly increased documentation of not only our software, but also our processes as well.

Robustness: Testing via Bamboo Continuous Integration

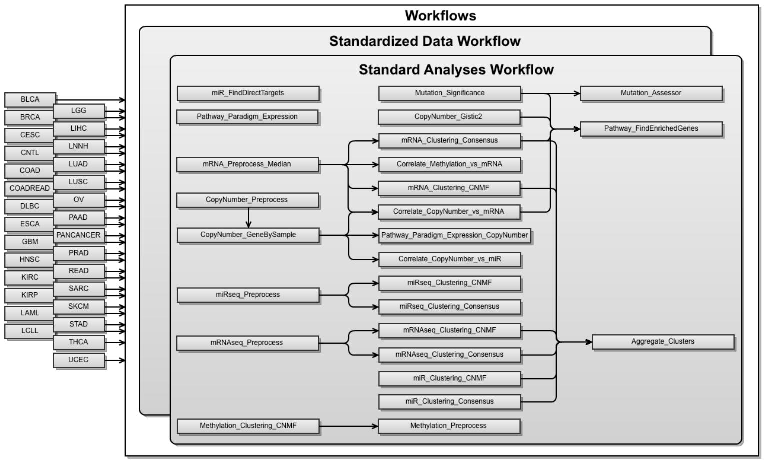


Plan	Build	Completed	Tests	Reason
GDAC Ingestor	✓ #260	3 hours ago	26 passed	Manual build by Daniel DiCara
Module Bam Realign BWA	✓ #75	5 months ago	10 passed	Dependant of CGA-NB-81
Module Create Merge Data Files SDRF	✓ #20	1 day ago	5 passed	Updated by Daniel DiCara
Module Iterative Scatter Gather Test	✓ #57	5 months ago	3 passed	Dependant of CGA-NB-81
Module PVCA Aggregator	✓ #14	4 months ago	3 passed	Updated by Daniel DiCara
Pipeline GISTIC 2	✓ #366	19 hours ago	21 passed	Updated by Chip Stewart
Pipeline Mutation Significance	! #374	19 hours ago	2 of 6 failed	Updated by Chip Stewart
Pipeline PARADIGM	✓ #172	1 month ago	19 passed	Manual build by Daniel DiCara
Pipeline PVCA	✓ #101	1 month ago	12 passed	Manual build by Daniel DiCara

Scalability: Parallelized Mirroring and Dicing



Scalability: Workflow Automation

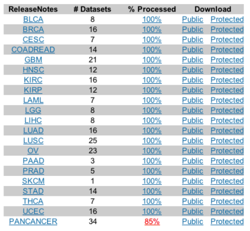


Robustness

Several process improvements have been made to improve the robustness of our work. First and foremost, a testing infrastructure has been developed. This infrastructure consists of a Bamboo Continuous Integration server and utilities for efficiently producing tests. It ensures that code under development is functioning properly and producing accurate results on a regular basis. And it allows our GDAC to quickly deploy algorithmic updates to our Analysis Pipeline in a dependable fashion. Second, our test coverage has increased significantly. As a consequence, it is simple to gauge the health of our GDAC pipelines via a dashboard in Bamboo that displays testing status. Third, logging has been significantly improved to facilitate reporting, diagnosing, and addressing bugs when they occur. Finally, an extensive effort to manually examine results to ensure their accuracy has been engrained in our week-to-week activities.

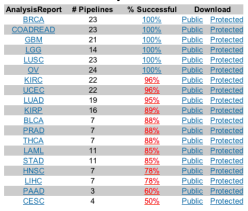
Transparency: Website & Dashboards

Standardized Data Dashboard



ReleaseNotes	# Datasets	% Processed	Download	
BLCA	8	100%	Public	Protected
BRCA	16	100%	Public	Protected
CESC	7	100%	Public	Protected
COADREAD	14	100%	Public	Protected
GBM	21	100%	Public	Protected
HNSC	12	100%	Public	Protected
KIRC	16	100%	Public	Protected
KIRP	12	100%	Public	Protected
LAML	7	100%	Public	Protected
LGG	8	100%	Public	Protected
LIHC	8	100%	Public	Protected
LUAD	16	100%	Public	Protected
LUSC	25	100%	Public	Protected
OV	23	100%	Public	Protected
PAAD	3	100%	Public	Protected
PRAD	5	100%	Public	Protected
SKCM	1	100%	Public	Protected
STAD	14	100%	Public	Protected
THCA	7	100%	Public	Protected
UCEC	16	100%	Public	Protected
PANCANCER	34	85%	Public	Protected

Standard Analyses Dashboard



AnalysisReport	# Pipelines	% Successful	Download	
BRCA	23	100%	Public	Protected
COADREAD	23	100%	Public	Protected
GBM	21	100%	Public	Protected
LGG	14	100%	Public	Protected
LUSC	23	100%	Public	Protected
OV	24	100%	Public	Protected
KIRC	22	96%	Public	Protected
UCEC	22	96%	Public	Protected
LUAD	19	95%	Public	Protected
KIRP	16	89%	Public	Protected
BLCA	7	88%	Public	Protected
PRAD	7	88%	Public	Protected
THCA	7	88%	Public	Protected
LAML	11	85%	Public	Protected
STAD	11	85%	Public	Protected
HNSC	7	78%	Public	Protected
LIHC	7	78%	Public	Protected
PAAD	3	60%	Public	Protected
CESC	4	50%	Public	Protected

Pinpoint Control and Extensibility

Pinpoint control has been enabled via improved Firehose APIs. This has allowed the quick and easy generation of scripts to modify, invoke, and monitor Firehose workspaces. It has also enabled the creation of dashboards on a regular basis to inform both internal and external users about the state of the Broad GDAC standardized data and analyses runs. In terms of extensibility, improved Firehose APIs have allowed us to extend the functionality of Firehose and rapidly prototype new features without modifying the Firehose source code. Our FIrhose Service Selector (FISS) tool is a good example. Repetitive operations and combinations of common API calls can be easily invoked via FISS. These use cases can ultimately inform and guide the development of new Firehose features.

Scalability

In terms of scalability, multiprocessing has been added in a number of places to make maximal use of compute resources. Furthermore, use of cron has enabled the scheduling of jobs at times when compute resources are typically under lighter load (i.e. nighttime and weekends). Finally, improving LSF usage by designing software that maximizes the efficiency of our farm has been a key priority.

Conclusions

The Broad GDAC has made substantial progress in improving the efficiency and reliability of our Firehose data ingestion, standardized data, and standard analyses activities. This was achieved by applying solid software engineering principals to not only our codes, but also to our processes. Our efforts focused on five key areas: clarity, transparency, robustness, pinpoint control and easy extensibility, and scalability. These improvements have facilitated the ingestion and processing of terabytes of data on a regular basis with minimal human interaction. Furthermore, they have improved our ability to add new data types and algorithms to our workflows in a quick and reliable fashion. Finally, these enhancements have allowed us to keep pace with the constantly increasing volume of data.