Butler enables rapid cloud-based analysis of thousands of human genomes

Originally published in:
Nature Biotechnology 38(3), https://doi.org/10.1038/s41587-019-0360-3

BRIEF COMMUNICATION
https://doi.org/10.1038/s41587-019-0360-3

nature biotechnology

There are amendments to this paper

OPEN

Butler enables rapid cloud-based analysis of thousands of human genomes

Sergei Yakneen[1,2,6]✉, Sebastian M. Waszak[1], PCAWG Technical Working Group[3], Michael Gertz[2], Jan O. Korbel[1,4]✉ and PCAWG Consortium[5]

We present Butler, a computational tool that facilitates large-scale genomic analyses on public and academic clouds. Butler includes innovative anomaly detection and self-healing functions that improve the efficiency of data processing and analysis by 43% compared with current approaches. Butler enabled processing of a 725-terabyte cancer genome dataset from the Pan-Cancer Analysis of Whole Genomes (PCAWG) project in a time-efficient and uniform manner.

Cloud computing offers easy and economical access to computational capacity at a scale that had previously been available to only the largest research institutions. To take advantage, large biological datasets are increasingly analyzed on various cloud computing platforms, using public, private and hybrid clouds[1] with the aid of workflow systems. When employed in global projects, such systems must be flexible in their ability to operate in different environments, including academic clouds, to allow researchers to bring their computational pipelines to the data, especially in cases where the raw data themselves cannot be moved. The recently developed cloud-based scientific workflow frameworks Nextflow[2], Toil[3] and GenomeVIP[4] focus their support largely on individual commercial cloud computing environments—mostly Amazon Web Services—and lack complete functionality for other major providers. This limits their use in studies that require multi-cloud operation due to practical and regulatory requirements[5,6]. Butler, in contrast, provides full support for operation on OpenStack-based commercial and academic clouds, Amazon Web Services, Microsoft Azure and Google Compute Platform, and can thus enable international collaborations involving the analysis of hundreds of thousands of samples where distributed cloud-based computation is pursued in different jurisdictions[5–7].

A key lesson learned from large-scale projects including the PCAWG project[7], which has pursued a study of 2,658 cancer genomes sequenced by the International Cancer Genome Consortium and the Cancer Genome Atlas, is that analysis of biological data of heterogeneous quality, generated at multiple locations with varying standard operating procedures, frequently suffers from artifacts that lead to many failures of computational jobs and that can considerably limit a project's progress. Sequencing library artifacts, sample contamination and nonuniform sequencing coverage[8] can cause data and software anomalies that challenge current workflows. Delays in recognizing and resolving these failures can notably affect data processing rate and increase project duration and costs. In contrast to previous tools, Butler provides an operational management toolkit that quickly discovers and resolves expected and unexpected failures (Fig. 1a,b and Supplementary Note 1).

The toolkit functions at two levels of granularity: host level and application level. Host-level operational management is facilitated via a health metrics system that collects system measurements at regular intervals from all deployed virtual machines (VMs). These metrics are aggregated and stored in a time-series database within Butler's monitoring server. A set of graphical dashboards reports system health to users while supporting advanced querying capabilities for in-depth troubleshooting (Supplementary Fig. 8). Application-level monitoring is facilitated via systematic log collection (Supplementary Fig. 4) and extraction wherein the logs are stored in a queryable search index[9]. These tools provide multidimensional visibility into operational bottlenecks and error conditions as they occur, in a manner that is aggregated across hundreds of VMs. On top of these data, a rule-based anomaly detection engine defines normal operating conditions that, when breached, trigger handling routines that can notify the user by sending e-mail, Slack or Telegram messages, and enables automated restarting of offending workflows, underlying services or entire VMs, allowing the cluster to self-heal (Fig. 1b).

These monitoring and operational management capabilities set Butler apart from current scientific workflow frameworks[2–4,10] (Supplementary Table 1), which do not contain anomaly detection modules and are therefore unable to automatically resolve key issues that frequently occur during large-scale analyses. For example, Butler's operational modules are able to identify and resolve failures of the cloud workflow scheduler, workflows that run perpetually and never finish (indicative of underlying problems), and crashed and unresponsive VMs that, in practice, may prevent workflows from setting a failed status and thus would prevent triggering of error handling logic in other workflow systems.

These capabilities indeed enable highly efficient data processing in studies, such as PCAWG, where analyses are run by multiple groups at different times and on different clouds. Butler can invoke a variety of analysis algorithms, including genome alignment, variant calling and execution of R scripts. These can either be preinstalled or run as Docker[11] images or Common Workflow Language (CWL)[12] tools and workflows. Butler's workflows accept parameters via JavaScript Object Notation (JSON) configuration files, which are stored in a database to maintain reproducibility. Workflow tasks scheduled for execution are deposited into a distributed task queue from which available worker nodes will pick them up, allowing analyses to be distributed over thousands of computing nodes. It is worth noting that for some small-scale projects executed over relatively short timelines, the increased complexity of setting up and

[1]European Molecular Biology Laboratory (EMBL), Genome Biology Unit, Heidelberg, Germany. [2]Institute of Computer Science, Heidelberg University, Heidelberg, Germany. [3]A list of members and affiliations appears at the end of the paper. [4]EMBL, European Bioinformatics Institute (EMBL-EBI), Hinxton, UK. [5]A list of members and affiliations appears in Supplementary Note 2. [6]Present address: Sophia Genetics SA, Saint Sulpice, Switzerland. ✉e-mail: llevar@gmail.com; jan.korbel@embl.de

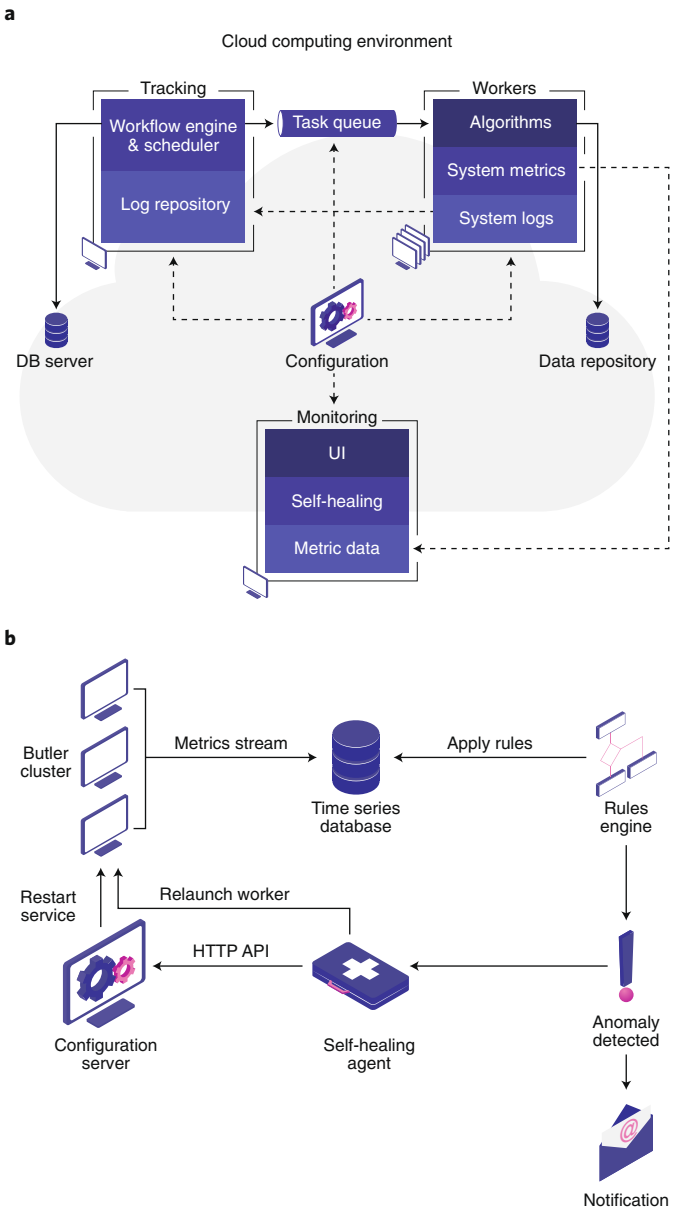

Fig. 1 | Butler framework architecture. a, The framework consists of several interconnected components, each running on a separate virtual machine (VM). See Methods and Supplementary Note 1 for details. **b**, Metrics flow from all VMs into a time series database. The self-healing agent detects anomalies and takes appropriate action. See Supplementary Note 1 for details. Solid arrows indicate information flow; dashed arrows indicate metrics flow; dashed-and-dotted arrows indicate configuration instructions.

running these monitoring systems may render Butler less practicable than simpler workflows.

We assessed Butler's ability to facilitate large-scale analyses of patient genomes in the context of the PCAWG study, where Butler was deployed on 1,500 CPU cores, 5.5 terabytes of random access memory (RAM), 1 petabyte of shared storage and 40 terabytes of local solid-state drive storage. Using Butler, we implemented and successfully tested a genomic alignment workflow using BWA[13], germline variant calling workflows based on FreeBayes[14] (Supplementary Fig. 5) and Delly[15], as well as several tools for somatic mutation calling, including Pindel[16] and BRASS[17]. We carried out whole-genome variant discovery and joint genotyping of 90 million germline genetic variants (single nucleotide polymorphisms (SNPs), indels and structural variants) across a 725-terabyte dataset comprising the full PCAWG cohort (including samples that were later blacklisted) of 2,834 cancer patients[7]. Additionally, we performed sequence alignment and called both germline and somatic variants on 232 high-coverage prostate cancer tumor–normal sample pairs in the context of the PanProstate Cancer Group (PPCG) Consortium. We executed and successfully completed over 2.5 million computational jobs using 546,552 CPU hours. The management overhead of employing Butler for these analyses was less than 2% of the overall computational cost.

To assess Butler performance in the field, in comparison to other large-scale workflow systems, we compare the actually observed historical performance of Butler, recorded during PCAWG, against the performance of the 'core' somatic PCAWG consortium pipelines (Fig. 2), which represent the current state of the art in the field in terms of cloud software[7] (on the basis of recency of development, scale of deployment, dataset size and analysis duration)—achieving nearly complete feature parity with several available cloud-based scientific workflow frameworks[2–4,10] (Supplementary Table 1). These PCAWG pipelines used the same information technology infrastructure and computed over the same samples, but did not use Butler. Our metric to estimate the highest achievable processing rate for an analysis is defined as the smallest proportion of time required for processing 5% of all samples, which we refer to as the 'target processing rate'. This is measured on the basis of the difference between the calendar completion date and time of the samples and the analysis start date, thus taking into account the time spent on failed and repeated runs and cluster downtime, which are major contributors to analysis duration. To establish how well a pipeline performs compared to its potential, we calculated the ratio of the actual processing rate to the target processing rate (Fig. 2a,b). Butler-operated pipelines were markedly closer to the target processing rate (mean actual/target rate ratio 0.696) than the core PCAWG pipelines (mean actual/target rate ratio 0.490) (Fig. 2c). Consequently, Butler-based analyses showed a duration 1.43 times the ideal target duration while core PCAWG pipelines showed a duration of 2.04 times the ideal target duration—43% longer. Additionally, core PCAWG pipelines exhibited a highly nonuniform processing rate (Fig. 2d) deviating 23.1% on average (minimum 0.0%, maximum 57.8%, s.d. 15.0%) from the ideally uniform trajectory of processing 1% of samples in 1% of analysis time, while Butler-based pipelines (Fig. 2e) performed in a substantially more uniform manner, deviating only 4.0% (minimum 0.0%, maximum 15.6%, s.d. 3.7%) over the same sample set on average (Methods). These timesaving and controlled execution abilities resulted in the adoption of Butler for genomics-oriented analyses in the context of the European Open Science Cloud (EOSC) Pilot (http://eoscpilot.eu) and its further adoption within PPGC (http://melbournebioinformatics.org.au/project/ppgc).

Butler can be generally applied to any large-scale analysis and could, for example, readily extend to studies such as GTEx (http://gtexportal.org), ENCODE (http://encodeproject.org) and the Human Cell Atlas Project (http://humancellatlas.org). A standard Butler workflow generically parallelizes R script execution across thousands of VMs, which will facilitate its use for other research contexts and other data types (including single-cell 'omics' data and microbiomes, for example).

We have developed Butler to meet the challenges of working with diverse cloud computing environments in the context of large-scale scientific data analyses. The operational management tools provided with Butler help overcome the key challenge that impacts analysis duration—the ability to autonomously detect, diagnose and address issues in a timely manner—thus allowing researchers to spend less time focusing on error conditions and considerably reduce analysis duration and cost. The comprehensive nature of the Butler toolkit sets it apart from current scientific workflow managers[2–4,10] (Supplementary Table 1) by offering an efficient and scalable solution for modern global cloud-based big data analyses.

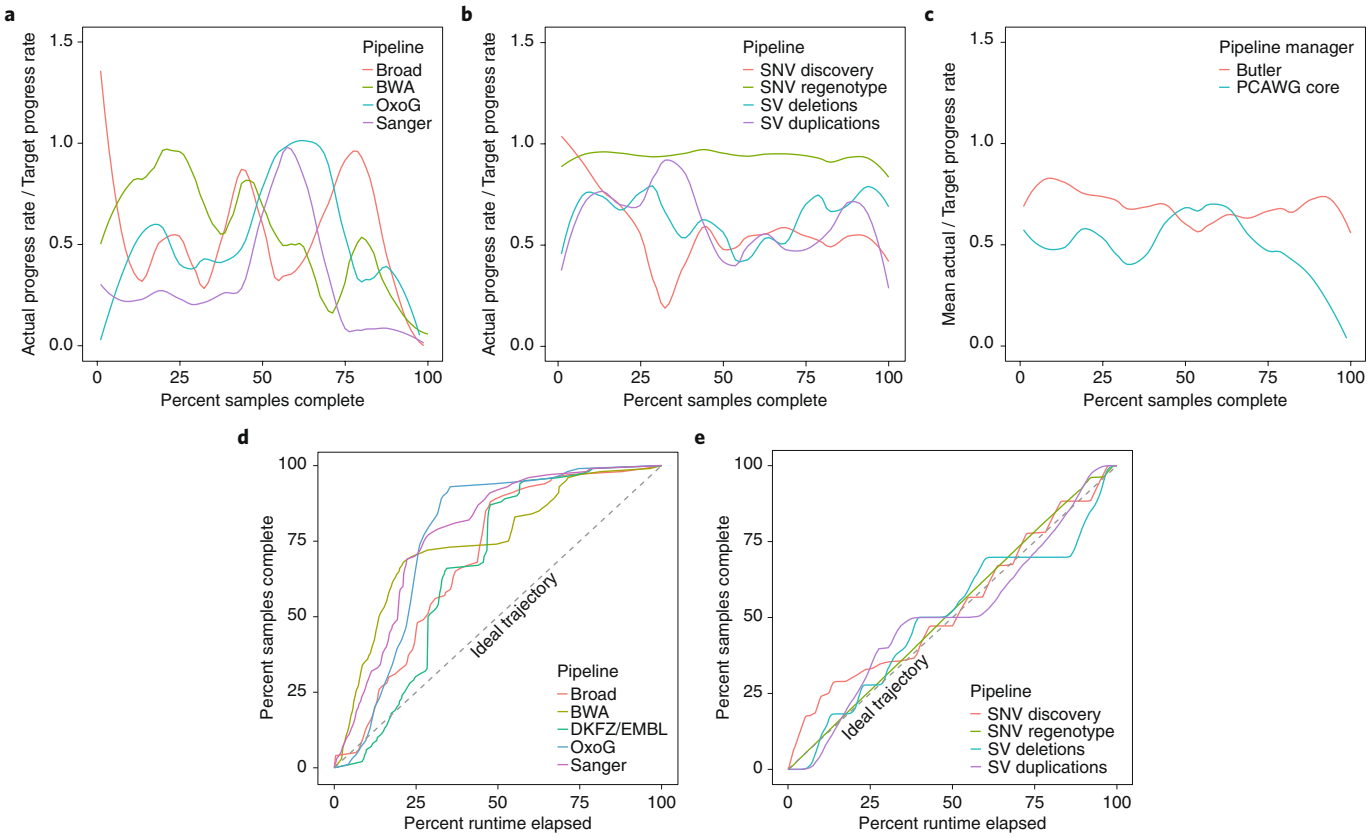

Fig. 2 | Butler performance comparison. a,**b**, Comparing the ratio of actual to target progress rates for core PCAWG pipelines (**a**) vs. Butler pipelines (**b**). See Methods for details. **c**, Mean actual/target progress rate ratio across pipelines for core PCAWG (mean 0.49) vs. Butler (mean 0.7) pipelines, each of which were run once over the entirety of PCAWG samples available to us. **d**,**e**, Progress rate uniformity of core PCAWG pipelines (**d**) vs. Butler (**e**). See Methods for details. In all panels the samples are arranged by their completion date. Runtime includes time spent on failed attempts. Comparison between Butler and core pipelines was facilitated in the context of the PCAWG. Similar comparison between Butler and other frameworks is presently impractical at this scale due to the high costs and complexity involved.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-019-0360-3.

Received: 14 July 2017; Accepted: 5 July 2018;
Published online: 5 February 2020

References

1. Habermann, N., Mardin, B. R., Yakneen, S. & Korbel, J. O. Using large-scale genome variation cohorts to decipher the molecular mechanism of cancer. *C. R. Biol.* **339**, 308–313 (2016).
2. Di Tommaso, P. et al. Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316–319 (2017).
3. Vivian, J. & Paten, B. Toil enables reproducible, open source, big biomedical data analyses. *Nat. Biotechnol.* **35**, 314–316 (2017).
4. Mashl, R. J. et al. GenomeVIP: a cloud platform for genomic variant discovery and interpretation. *Genome Res.* **27**, 1450–1459 (2017).
5. Stein, L. D., Knoppers, B. M., Campbell, P., Getz, G. & Korbel, J. O. Data analysis: create a cloud commons. *Nature* **523**, 149–151 (2015).
6. Molnár-Gábor, F., Lueck, R., Yakneen, S. & Korbel, J. O. Computing patient data in the cloud: practical and legal considerations for genetics and genomics research in Europe and internationally. *Genome Med.* **9**, 58 (2017).
7. Pan-cancer Analysis of Whole Genomes Consortium. Pan-cancer analysis of whole genomes. *Nature* https://doi.org/10.1038/s41586-020-1969-6 (2020).
8. Soergel, D. A. Rampant software errors may undermine scientific results. *F1000 Res.* **3**, 303 (2014).
9. Gormley, C. & Tong, Z. *Elasticsearch: The Definitive Guide* (O'Reilly Media, 2015).
10. Leipzig, J. A review of bioinformatic pipeline frameworks. *Brief. Bioinformatics* **18**, 530–536 (2017).
11. Merkel, D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* **2014**, 2 (2014).
12. Amstutz, P. et al. Common Workflow Language, v1. 0. https://w3id.org/cwl/v1.0/; https://doi.org/10.6084/m9.figshare.3115156.v2 (2016).
13. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
14. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. Preprint at https://arxiv.org/abs/1207.3907 (2012).
15. Rausch, T. et al. DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28**, i333–i339 (2012).
16. Raine, K. M. et al. cgpPindel: identifying somatically acquired insertion and deletion events from paired end sequencing. *Curr. Protoc. Bioinformatics* **15**, 15.7.11–15.7.12 (2015).
17. Nik-Zainal, S. et al. Landscape of somatic mutations in 560 breast cancer whole-genome sequences. *Nature* **534**, 47–54 (2016).

PCAWG Technical Working Group

Sergei Yakneen[1,2,6], Brice Aminou[7], Javier Bartolome[8], Keith A. Boroevich[9,10], Rich Boyce[4], Angela N. Brooks[11,12,13], Alex Buchanan[14], Ivo Buchhalter[15,16,17], Adam P. Butler[18], Niall J. Byrne[7], Andy Cafferkey[4], Peter J. Campbell[18,19], Zhaohong Chen[20], Sunghoon Cho[21], Wan Choi[22], Peter Clapham[18], Brandi N. Davis-Dusenbery[23], Francisco M. De La Vega[24,25,26,27], Jonas Demeulemeester[28,29], Michelle T. Dow[20], Lewis Jonathan Dursi[30,31], Juergen Eils[32,33], Roland Eils[15,17,32,33], Kyle Ellrott[14], Claudiu Farcas[20], Francesco Favero[34], Nodirjon Fayzullaev[7], Vincent Ferretti[7,35], Paul Flicek[4], Nuno A. Fonseca[4,36], Josep Ll. Gelpi[8,37], Gad Getz[11,38,39,40], Bob Gibson[7], Robert L. Grossman[41], Olivier Harismendy[42], Allison P. Heath[43], Michael C. Heinold[15,17], Julian M. Hess[11,44], Oliver Hofmann[45], Jongwhi H. Hong[46], Thomas J. Hudson[47,48], Barbara Hutter[49,50,51], Carolyn M. Hutter[52], Daniel Hübschmann[17,32,53,54,55], Seiya Imoto[56], Sinisa Ivkovic[57], Seung-Hyup Jeon[22], Wei Jiao[30], Jongsun Jung[58], Rolf Kabbe[15], Andre Kahles[59,60,61,62,63], Jules N. A. Kerssemakers[15], Hyung-Lae Kim[64], Hyunghwan Kim[22], Jihoon Kim[65], Youngwook Kim[66,67], Kortine Kleinheinz[15,17], Michael Koscher[68], Antonios Koures[20], Milena Kovacevic[58], Chris Lawerenz[33], Ignaty Leshchiner[11], Jia Liu[69], Dimitri Livitz[11], George L. Mihaiescu[7], Sanja Mijalkovic[57], Ana Mijalkovic Lazic[57], Satoru Miyano[56], Naoki Miyoshi[56], Hardeep K. Nahal-Bose[7], Hidewaki Nakagawa[10], Mia Nastic[57], Steven J. Newhouse[4], Jonathan Nicholson[18], Brian D. O'Connor[7,13], David Ocana[4], Kazuhiro Ohi[56], Lucila Ohno-Machado[20], Larsson Omberg[70], B. F Francis Ouellette[71,72], Nagarajan Paramasivam[15,50], Marc D. Perry[7,73], Todd D. Pihl[74], Manuel Prinz[15], Montserrat Puiggròs[75], Petar Radovic[57], Keiran M. Raine[18], Esther Rheinbay[11,40,76], Mara Rosenberg[11,76], Romina Royo[75], Gunnar Rätsch[59,62,63,77,78,79], Gordon Saksena[11], Matthias Schlesner[15,80], Solomon I. Shorser[30], Charles Short[4], Heidi J. Sofia[52], Jonathan Spring[41], Lincoln D. Stein[30,81], Adam J. Struck[14], Grace Tiao[11], Nebojsa Tijanic[57], David Torrents[75,82], Peter Van Loo[28,29], Miguel Vazquez[76,82], David Vicente[75], Jeremiah A. Wala[11,40,12], Zhining Wang[83], Sebastian M. Waszak[1], Joachim Weischenfeldt[1,84,85], Johannes Werner[15,86], Ashley Williams[20], Youngchoon Woo[22], Adam J. Wright[30], Qian Xiang[87], Liming Yang[83], Denis Yuen[30], Christina K. Yung[7], Junjun Zhang[7] and Jan O. Korbel[1,4]

[7]Genome Informatics Program, Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [8]Barcelona Supercomputing Center (BSC), Barcelona, Spain. [9]Laboratory for Medical Science Mathematics, RIKEN Center for Integrative Medical Sciences, Yokohama, Kanagawa, Japan. [10]RIKEN Center for Integrative Medical Sciences, Yokohama, Kanagawa, Japan. [11]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [12]Dana-Farber Cancer Institute, Boston, MA, USA. [13]University of California Santa Cruz, Santa Cruz, CA, USA. [14]Oregon Health and Science University, Portland, OR, USA. [15]Division of Theoretical Bioinformatics, German Cancer Research Center (DKFZ), Heidelberg, Germany. [16]Heidelberg Center for Personalized Oncology (DKFZ-HIPO), German Cancer Research Center, Heidelberg, Germany. [17]Institute of Pharmacy and Molecular Biotechnology and BioQuant, Heidelberg University, Heidelberg, Germany. [18]Wellcome Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge, UK. [19]Department of Haematology, University of Cambridge, Cambridge, UK. [20]University of California San Diego, San Diego, CA, USA. [21]PDXen Biosystems Inc, Seoul, South Korea. [22]Electronics and Telecommunications Research Institute, Daejeon, South Korea. [23]Seven Bridges Genomics, Charlestown, MA, USA. [24]Annai Systems, Inc, Carlsbad, CA, USA. [25]Department of Biomedical Data Science, Stanford University School of Medicine, Stanford, CA, USA. [26]Department of Genetics, Stanford University School of Medicine, Stanford, CA, USA. [27]Departments of Genetics and Biomedical Data Science, Stanford University School of Medicine, Stanford, CA, USA. [28]University of Leuven, Leuven, Belgium. [29]The Francis Crick Institute, London, UK. [30]Computational Biology Program, Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [31]The Hospital for Sick Children, Toronto, Ontario, Canada. [32]Heidelberg University, Heidelberg, Germany. [33]New BIH Digital Health Center, Berlin Institute of Health (BIH) and Charité – Universitätsmedizin Berlin, Berlin, Germany. [34]Rigshospitalet, Copenhagen, Denmark. [35]Department of Biochemistry and Molecular Medicine, University of Montreal, Montreal, Quebec, Canada. [36]CIBIO/InBIO— Research Center in Biodiversity and Genetic Resources, Universidade do Porto, Vairão, Portugal. [37]Department Biochemistry and Molecular Biomedicine, University of Barcelona, Barcelona, Spain. [38]Center for Cancer Research, Massachusetts General Hospital, Boston, MA, USA. [39]Department of Pathology, Massachusetts General Hospital, Boston, MA, USA. [40]Harvard Medical School, Boston, MA, USA. [41]University of Chicago, Chicago, IL, USA. [42]Division of Biomedical Informatics, Department of Medicine, & Moores Cancer Center, UC San Diego School of Medicine, San Diego, CA, USA. [43]Children's Hospital of Philadelphia, Philadelphia, PA, USA. [44]Massachusetts General Hospital Center for Cancer Research, Charlestown, MA, USA. [45]University of Melbourne Centre for Cancer Research, University of Melbourne, Melbourne, Victoria, Australia. [46]Syntekabio Inc, Daejeon, South Korea. [47]AbbVie, North Chicago, IL, USA. [48]Genomics Program, Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [49]German Cancer Consortium (DKTK), Heidelberg, Germany. [50]Heidelberg Center for Personalized Oncology (DKFZ-HIPO), German Cancer Research Center (DKFZ), Heidelberg, Germany. [51]National Center for Tumor Diseases (NCT) Heidelberg, Heidelberg, Germany. [52]National Human Genome Research Institute, National Institutes of Health, Bethesda, MD, USA.

[53]Department of Pediatric Immunology, Hematology and Oncology, University Hospital, Heidelberg, Germany. [54]German Cancer Research Center (DKFZ), Heidelberg, Germany. [55]Heidelberg Institute for Stem Cell Technology and Experimental Medicine (HI-STEM), Heidelberg, Germany. [56]Institute of Medical Science, University of Tokyo, Tokyo, Japan. [57]Seven Bridges, Charlestown, MA, USA. [58]Genome Integration Data Center, Syntekabio, Inc, Daejeon, South Korea. [59]Computational Biology Center, Memorial Sloan Kettering Cancer Center, New York, NY, USA. [60]ETH Zurich, Department of Biology, Zurich, Switzerland. [61]ETH Zurich, Department of Computer Science, Zurich, Switzerland. [62]SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland. [63]University Hospital Zurich, Zurich, Switzerland. [64]Department of Biochemistry, College of Medicine, Ewha Womans University, Seoul, South Korea. [65]Health Sciences Department of Biomedical Informatics, University of California San Diego, La Jolla, CA, USA. [66]Department of Health Sciences and Technology, Sungkyunkwan University School of Medicine, Seoul, South Korea. [67]Samsung Genome Institute, Seoul, South Korea. [68]Functional and Structural Genomics, German Cancer Research Center (DKFZ), Heidelberg, Germany. [69]Leidos Biomedical Research, Inc, McLean, VA, USA. [70]Sage Bionetworks, Seattle, WA, USA. [71]Genome Informatics, Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [72]Department of Cell and Systems Biology, University of Toronto, Toronto, Ontario, Canada. [73]Department of Radiation Oncology, University of California San Francisco, San Francisco, CA, USA. [74]CSRA Incorporated, Fairfax, VA, USA. [75]Barcelona Supercomputing Center, Barcelona, Spain. [76]Massachusetts General Hospital, Boston, MA, USA. [77]Department of Biology, ETH Zurich, Zurich, Switzerland. [78]Department of Computer Science, ETH Zurich, Zurich, Switzerland. [79]Weill Cornell Medical College, New York, NY, USA. [80]Bioinformatics and Omics Data Analytics, German Cancer Research Center (DKFZ), Heidelberg, Germany. [81]Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada. [82]Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain. [83]National Cancer Institute, National Institutes of Health, Bethesda, MD, USA. [84]Finsen Laboratory and Biotech Research & Innovation Centre (BRIC), University of Copenhagen, Copenhagen, Denmark. [85]Department of Urology, Charité Universitätsmedizin Berlin, Berlin, Germany. [86]Department of Biological Oceanography, Leibniz Institute of Baltic Sea Research, Rostock, Germany. [87]Ontario Institute for Cancer Research, Toronto, Ontario, Canada.

Methods

The Butler system. Overall, the Butler system is composed of four distinct subsystems. The Cluster Lifecycle Management is the first subsystem and deals with the task of creating and tearing down clusters on various clouds, including defining VMs, storage devices, network topology and network security rules. The second subsystem, Cluster Configuration Management, deals with configuration and software installation of all VMs in the cluster. The Workflow System is responsible for allowing users to define and run scientific workflows on the cloud. Finally, the Operational Management subsystem provides tools for ensuring continuous successful operation of the cluster, as well as for troubleshooting error conditions. Supplementary Note 1 contains an in-depth description of each of these subsystems and how they work within Butler, while the Installation Guide (http://butler.readthedocs.io/en/latest/installation.html) provides detailed instructions for how to set up the software.

Butler deployment. Butler has been validated for production use on the EMBL-EBI Embassy Cloud (http://www.embassycloud.org), an academic cloud computing center that runs an OpenStack-based environment (Fig. 1). The Embassy Cloud has played a key role in the PCAWG project by donating substantial storage and cloud computing capacity over the course of 3 years. The total amount of resources dedicated to the project by the Embassy Cloud was as follows:

- 1 PB Isilon storage shared over NFS
- 1,500 computational cores
- 5.5 TB RAM
- 40 TB local solid-state drive storage
- 10-gigabit network

These resources have been used to host one of the six PCAWG data repositories that exist worldwide, as well as performing scientific analyses for the project. We have used Butler extensively on the Embassy Cloud to carry out the analyses for the PCAWG Germline Working Group. To deploy Butler on the 1,500-core cluster, we set up five different profiles of VMs, each playing several different roles (Supplementary Table 2).

Each profile was defined separately via Terraform and uses Saltstack roles for configuration. Users can check out the Butler github repository to their local machine, and once they install Terraform locally, they can fully commandeer the provisioning process from the local machine via Terraform.

The cluster is bootstrapped via the Salt-master VM. This VM is started first whenever the cluster needs to be recreated from scratch. The monitoring-server role is responsible for installing and configuring InfluxDB and other monitoring components, as well as registering them with Consul so that metrics can start being recorded. We also attach a 1-TB block storage volume for the metrics database so that it can survive cluster crashes and teardowns. If the monitoring server needs to be recreated, the block storage volume simply needs to be reattached to the new Monitoring Server VM.

The tracker VM is responsible for running various Airflow components, such as the Scheduler, Webserver and Flower. Additionally, we deploy the Butler tracker module to this VM, and thus the tracker VM acts as the main control point of the system from which analyses are launched and monitored. This VM additionally has the Elasticsearch role that designates it as the location of the Logstash and Elasticsearch components. To persist the search index, we attach an additional 1-TB block storage volume.

The job queue VM is responsible for hosting the RabbitMQ server, which holds all of the in-flight workflow tasks. Because the resources of the job queue are heavily taxed by communication with all of the worker VMs in the cluster, we do not assign any additional roles to this host.

The db-server is responsible for hosting most of the databases used by Butler. This VM runs an instance of PostgreSQL Server and hosts the Run Tracking DB, Airflow DB and Sample Tracking DB. The 1-TB block storage volume serves as the backing storage mechanism.

The worker VMs are the workhorses of the Butler cluster. For analyses by the PCAWG Germline Working Group, we employed 175 eight-core worker machines dedicated to running Butler workflows. The worker role ensures that Airflow client modules are installed and loaded on each worker. The germline role also loads the workflows and analyses that are relevant to the PCAWG Germline Working Group.

Because of the comprehensive nature of the Butler framework, which covers far more scope than a traditional workflow framework (provisioning, configuration management, operations management, anomaly detection, etc.), the setup and deployment of a Butler system are more complex than those of other workflow frameworks because multiple VMs need to be successfully set up and configured to interact with each other in a secure environment that is fit for sensitive information handling. Even though Butler features comprehensive documentation (http://butler.readthedocs.io), usage examples and automated deployment and configuration scripts, we recommend that the prospective user should ideally have a working understanding of cloud computing, server administration, networking, security, and other development operations (dev ops) concepts to make full use of the system. And while smaller-scale projects may benefit less from Butler's state-of-the-art feature set owing to its increased complexity and learning curve, this feature set is imperative for enabling the success of current and future generations of large-scale bioinformatics computing on the cloud.

PCAWG germline analyses. To assess Butler's performance on real data, we carried out several large-scale data analyses using Butler on the Embassy Cloud and over the entirety of the 725 TB of raw PCAWG data, including the following:

- discovery of germline single nucleotide variants (SNVs) and small indels in normal genomes.
- genotyping of common SNVs occurring at minor allele frequency (MAF) >1% in the 1000 Genomes Project[18].
- genotyping of germline SNVs and small indels in tumor and normal genomes (Supplementary Fig. 6).
- discovery and genotyping of structural variant deletions in tumor and normal genomes (Supplementary Fig. 7).
- discovery and genotyping of structural variant duplications in tumor and normal genomes (Supplementary Fig. 7).

Overall, most Butler workflows that carry out an analysis follow a similar structure (Supplementary Fig. 1): an analysis run is started, access to the sample is validated, the analysis steps are carried out (possibly with branching), and the analysis run is completed. Because of the largely common structure between workflows a large degree of code reuse is possible, and thus most of the methods reside in the workflow_common submodule of the Analysis Tracker and are invoked for each workflow.

Common variant genotyping was performed across the PCAWG cohort using a site list of 12 million variants occurring with at least 1% minor allele frequency within the 1000 Genomes Project[18] phase 3 cohort, interrogating 34 billion sites overall. 130,152 computing hours were used to complete 70,850 workflow tasks for this analysis, with an additional 2,688 CPU hours used for cluster management overhead. Thus, management overhead accounted for 2% of the overall computational resource costs for this analysis. Using 1,000 cores, this analysis took less than 6 d to complete. Supplementary Fig. 2 shows a distribution of job runtimes by chromosome (runtimes highly correlate with chromosome length, $r = 0.92$). Using a site list of 60 million variants obtained from the FreeBayes Variant Discovery analysis, we used the Butler FreeBayes Workflow in genotyping mode to calculate genotypes at 170 billion genomic positions. 76,518 workflow tasks were completed using 302,071 CPU hours over the course of the analysis (10 d wall time), of which 5,040 CPU hours were cluster management overhead, accounting for 1.6% of total resource utilization.

244,889 deletions were evaluated across 5,668 samples (tumor and normal) for a total of 1,388,030,852 genomic sites genotyped. Overall wall time was 13 d, using 265,200 CPU hours with 6,240 CPU hours going to cluster management overhead—an overhead of 2.2%. 217,433 duplications were genotyped for each sample across 5,668 samples, for a total of 1,232,410,244 genomic variants genotyped. The wall time for this analysis was only 4.5 d, using 151,200 CPU hours during this time, with a management overhead of 2,160 h, for a total overhead of 1.4%. The comparatively low cluster management overhead has been accomplished by scaling up the cluster to 1,400 cores without the need for more management resources. Supplementary Fig. 3 shows a distribution of workflow run durations.

We carried out several analyses on a 725-TB dataset of 2,834 cancer patients' genomic samples, consuming a total of 546,552 CPU hours. Each analysis took no longer than 2 weeks to complete and used only 1.5%–2.2% of the overall computing capacity for management overhead. On several occasions we were able detect large-scale cluster instability and program crashes using the Operational Management system and take corrective action with a minimal impact on overall productivity.

Comparing Butler with the core PCAWG somatic pipelines. We evaluate the relative effectiveness of Butler-based pipelines in comparison to a set of pipelines operating under similar conditions and over the same dataset, namely the 'core' PCAWG somatic pipelines that have been used to accomplish genome alignment and somatic variant calling for the PCAWG Technical Working Group[7]. The core PCAWG pipeline set consists of five pipelines—BWA, Sanger, Broad, DKFZ/EMBL and OxoG detection—run over the course of 18 months over all samples in PCAWG. The Butler-based pipeline set consists of two pipelines—FreeBayes and Delly, used to accomplish four analyses: germline SNV discovery, germline SNV genotyping, germline structural variant deletion genotyping and germline structural variant duplication genotyping—also running over all samples in PCAWG (725 TB in total). We assessed and compared pipeline performance with respect to an estimated optimal performance (based on available hardware), as well as with respect to analysis progress uniformity in time.

For core PCAWG pipelines, we used the date of data upload to the official data repository as the most reliable sample completion date. However, approximately 25% of the DKFZ/EMBL pipeline results were uploaded in two batches on two separate days, and thus do not accurately represent the real analysis progress rate. For this reason, we excluded this pipeline from the optimal performance analysis. Butler sample completion dates are based on timestamps collected in Butler's analysis tracking database.

Our assessment of pipeline performance is based on establishing an 'optimal' progress rate for a pipeline given a hardware allocation. We divided the sample set into 20 bins based on their completion time (each bin comprising 5% of all samples) and defined the optimal progress rate for each pipeline to be the smallest

proportion of overall analysis time required to process all samples of a bin (scaled to a 1% rate).

$$r_{\text{opt}} = \min_{b \in \text{bins}} \{\text{duration}_b / \text{duration}_{\text{total}} / 5\}$$

We observed that the mean r_{opt} was significantly higher for Butler-based pipelines at 0.46 than for the core PCAWG pipelines at 0.13 (Supplementary Table 3). For each pipeline and each 1% of the samples under analysis, we then computed a metric e (for effectiveness) defined as the proportion of r_{opt} actually achieved.

$$e = \frac{r_{\text{act}}}{r_{\text{opt}}}$$

Comparing the core PCAWG and Butler pipelines with respect to e (Fig. 2a–c), we observed that effectiveness was on average lower for PCAWG pipelines ($\mu_{e_{\text{PCAWG}}} = 0.49$) than for Butler pipelines ($\mu_{e_{\text{Butler}}} = 0.70$). Assessing the expected analysis duration for the two sets of pipelines, we observed

$$d_{\text{PCAWG}} = \frac{100}{\mu_{e_{\text{PCAWG}}}} = 2.04 d_{\text{opt}}$$

$$d_{\text{Butler}} = \frac{100}{\mu_{e_{\text{Butler}}}} = 1.43 d_{\text{opt}}$$

$$d_{\text{PCAWG}} = 1.43 d_{\text{Butler}}$$

Thus, the estimated duration for PCAWG pipelines was 43% longer than that for Butler-based pipelines.

We further compared core PCAWG pipelines with Butler pipelines on the basis of uniformity of rate of progress through an analysis. Given a constant resource allocation, an ideal analysis execution processes 1% of all samples in 1% of the analysis runtime. We divided the sample set into 100 equal-size bins and measured the percentage of overall analysis time spent processing each bin (Fig. 2d,e). Deviations from the diagonal indicate inefficiencies in data processing. Measuring this deviation, we observed that PCAWG pipelines deviated 23.1% from the diagonal on average (minimum 0.0%, maximum 57.8%, s.d. 15.0%) while Butler pipelines over the same sample set only deviated 4.0% (minimum 0.0%, maximum 15.6%, s.d. 3.7%) from the diagonal on average. This indicates that Butler pipelines are considerably less affected by various causes that slow an analysis (for example, job and infrastructure failures).

Adapting Butler to new projects and domains. Butler is a highly general workflow framework, built on top of generic open source components that in principle can work with any data in any scientific domain, deploy onto over 20 cloud types, and work on any operating system, and it comprises a rich set of tools for installing and configuring software. Adapting Butler to a new application is straightforward. This process is described below.

Butler has a prebuilt library of workflows that focus on handling genomic data and can support a large variety of studies that are based on next-generation sequencing applications, such as variant discovery, common and rare variant association studies, cancer genome analysis, and expression quantitative trait locus (eQTL) mapping. Using one of these workflows is simply a matter of providing configuration values in JSON format for the underlying tools (such as, for example, FreeBayes, Delly, samtools[19] or bcftools). Notably, Butler also supplies a generic workflow that allows execution of arbitrary R scripts across the entire Butler cluster. This powerful functionality can be used to facilitate a broad range of studies across disciplines, communities and analysis types, given the wide cross-community usage of R.

If the prebuilt workflows do not meet the users' requirements as-is, they can be customized to adapt to arbitrary needs or entirely new workflows can be written. Each Butler workflow is a Python program, which typically contains only 100–200 lines of code. There are three principal avenues of developing new workflows that are suitable to a wide variety of users' needs.

The easiest involves adapting tools that are already available as Docker images. Butler has prebuilt configurations for setting up all the infrastructure necessary to run Docker containers. The user only needs to wrap the Docker command line within existing boilerplate code that sets up access to the data that need to be analyzed. Once appropriate configuration parameters are supplied, Butler will be able to run the workflow seamlessly.

Only slightly more sophisticated is the setup of workflows that use CWL (Common Workflow Language) as a description language. Butler already has built-in functionality for installing and configuring cwl-runner, which is the reference implementation of CWL. To set up a new workflow that uses CWL within Butler, users need to prepare an appropriate JSON parameter file according to the CWL definition. This is accomplished via Butler's configuration functionality. The genome alignment and somatic variant calling workflows that accompany the Butler framework already provide full functionality in this regard and can be used as examples by new users. Because a number of workflows from varying scientific fields have already been described with CWL, this approach opens up a relatively straightforward avenue for adopting Butler in a wide variety of additional studies.

Potentially the most complex, but also the most powerful, way of authoring new workflows is writing them using the native constructs of the underlying Apache Airflow workflow framework. This approach provides the users with all of the power of the Python language and extended library, as well as the prebuilt Airflow components for interacting with a wide variety of distributed systems and engines, such as HDFS, Apache Spark, Apache Cassandra, various databases such as PostgreSQL and SQLite, email engines and many more. Several of the prebuilt Butler workflows, such as the FreeBayes, Delly and R workflow, use this approach, and users can employ these as templates for new workflows built in this style.

Because of the wide variety of workflow authoring and customization styles available, the existing examples, and the generic nature of the underlying open source components, applying Butler to new projects and analysis domains can be accomplished with minimal efforts and at a complexity level that is matched to the requirements of the project. Individual steps of the workflow can be easily debugged and tested on the local machine without the need to deploy to any cloud, using Python's extensive testing and debugging functionality. The typical life cycle for developing a new workflow is a few hours to a few days long and is usually much shorter than a week. Because new projects frequently require the installation and configuration of new software packages, Butler has integrated a full-featured configuration management solution called Saltstack that is used to set up and configure Butler internals and also any additional software required by the user for their project. Recipes for configuring dozens of software packages are already included with the Butler system, and hundreds more are available as community contributions to the Saltstack project. Arbitrary new configurations can be defined by the user to meet their custom requirements. To support this the user would typically set up a new Github repository that acts as a customization layer on top of the core Butler configurations. Within this custom repository, users can define new configuration recipes or override the behavior of the pre-existing Butler settings depending on the needs of their scientific project. We provide several examples of such repositories under 'Code availability' to help users become familiar with Butler.

Statistics. No formal sample size and power calculations were performed as we made use of all 5,668 of the samples available to us via the PCAWG consortium. The analyses in Fig. 2, performed over the entirety of PCAWG samples available to us, were run once (rather than multiple times) owing to the multi-year nature and high costs of the PCAWG project.

Ethical compliance. The authors have complied with all of the relevant ethical regulations with regards to the subjects described in this manuscript.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

PCAWG's final callsets, somatic and germline variant calls, mutational signatures, subclonal reconstructions, transcript abundance, splice calls and other core data generated by the ICGC/TCGA Pan-cancer Analysis of Whole Genomes Consortium is described in ref. [7] and available for download at https://dcc.icgc.org/releases/PCAWG. Additional information on accessing the data, including raw read files, can be found at https://docs.icgc.org/pcawg/data/. In accordance with the data access policies of the ICGC and TCGA projects, most molecular, clinical and specimen data are in an open tier that does not require access approval. To access potentially identifying information, such as germline alleles and underlying sequencing data, researchers will need to apply to the TCGA Data Access Committee (DAC) via dbGaP (https://dbgap.ncbi.nlm.nih.gov/aa/wga.cgi?page=login) for access to the TCGA portion of the dataset and to the ICGC Data Access Compliance Office (DACO; http://icgc.org/daco) for access to the ICGC portion. In addition, to access somatic single nucleotide variants derived from TCGA donors, researchers will also need to obtain dbGaP authorization.

Code availability

The source code for Butler is freely available at http://github.com/llevar/butler under the GPL v3.0 license.

The project-specific deployment settings, configurations, analysis definitions, and workflows are available at the following:

PCAWG Germline Project: https://github.com/llevar/pcawg-germline

EOSC Pilot: https://github.com/llevar/eosc_pilot

Pan-Prostate Cancer Group: https://github.com/llevar/pan-prostate

The R source code for the analysis is available at https://github.com/llevar/butler_perf_analysis.

The core computational pipelines used by the PCAWG Consortium for alignment, quality control and variant calling are available to the public at https://dockstore.org/search?search=pcawg under the GNU General Public License v3.0, which allows for reuse and distribution.

References
18. Auton, A. et al. A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).
19. Li, H. et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).

Acknowledgements
We acknowledge the contributions of the many clinical networks across ICGC and TCGA who provided samples and data to the PCAWG Consortium, and the contributions of the Technical Working Group and the Germline Working Group of the PCAWG Consortium for collation, realignment and harmonized variant calling of the PCAWG cancer genomes. We thank the patients and their families for their participation in the individual ICGC and TCGA projects. We also thank the PPCG project, and J. Weischenfeldt for assistance with the PPCG data. We are grateful to C. Yung, B. O'Connor, J. Zhang and L. Stein for their assistance and invaluable advice throughout the project and to A. Cafferkey, C. Short, D. Ocaña, D. Vianello, E. van den Bergh, S. Newhouse and E. Birney for invaluable support with the EMBL-EBI Embassy Cloud used largely for the computing in this study. We also acknowledge The Cancer Genome Collaboratory, Amazon Web Services, Google Compute Platform and Microsoft Azure for providing computing or cloud infrastructure. J.O.K. acknowledges support by the EOSC Pilot study (European Commission award number 739563), the BMBF (de.NBI project 031A537B), the European Research Council (336045) and the Heidelberg Academy of Sciences and Humanities. S.W. was supported through an SNSF Early Postdoc Mobility fellowship (P2ELP3_155365) and an EMBO Long-Term Fellowship (ALTF 755-2014).

Author contributions
This manuscript was written by S.Y. and J.O.K., with input from all authors. S.Y. and J.O.K. are responsible for study conception. S.Y. designed, implemented, and executed the Butler software framework in the context of the analyses described in this manuscript. S.M.W. designed workflows and assessed the integrity of the framework. S.Y. led the data analysis, and S.M.W., M.G. and J.O.K. contributed to data analysis. The PCAWG Technical Working group provided invaluable assistance and feedback. M.G. and J.O.K. provided supervision and project oversight.

Competing interests
G.G. receives research funds from IBM and Pharmacyclics and is an inventor on patent applications related to MuTect, ABSOLUTE, MutSig, MSMuTect, MSMutSig and POLYSOLVER.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41587-019-0360-3.

Correspondence and requests for materials should be addressed to S.Y. or J.O.K.

Reprints and permissions information is available at www.nature.com/reprints.

nature research

Corresponding author(s): Jan Korbel, Sergei Yakneen

Life Sciences Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form is intended for publication with all accepted life science papers and provides structure for consistency and transparency in reporting. Every life science submission will use this form; some list items might not apply to an individual manuscript, but all fields must be completed for clarity.

For further information on the points included in this form, see Reporting Life Sciences Research. For further information on Nature Research policies, including our data availability policy, see Authors & Referees and the Editorial Policy Checklist.

Please do not complete any field with "not applicable" or n/a. Refer to the help text for what text to use if an item is not relevant to your study.
For final submission: please carefully check your responses for accuracy; you will not be able to make changes later.

Experimental design

1. Sample size

   Describe how sample size was determined.

   The sample size corresponds to all whole cancer genomes that at the time of the commencement of the Pan-Cancer Analysis of Whole Genomes (PCAWG) study had been completed by deep massively parallel sequencing within the International Cancer Genome Consortium (ICGC) and the Cancer Genome Atlas (TCGA).

2. Data exclusions

   Describe any data exclusions.

   No data were excluded.

3. Replication

   Describe the measures taken to verify the reproducibility of the experimental findings.

   Not applicable. We analyzed all data available, namely, all whole cancer genomes that at the time of the commencement of the Pan-Cancer Analysis of Whole Genomes (PCAWG) study had been completed by deep massively parallel sequencing, by the ICGC and the TCGA.

4. Randomization

   Describe how samples/organisms/participants were allocated into experimental groups.

   No randomization was necessary. We analyzed all data available, namely, all whole cancer genomes that at the time of the commencement of the Pan-Cancer Analysis of Whole Genomes (PCAWG) study had been completed by deep massively parallel sequencing, by the ICGC and the TCGA.

5. Blinding

   Describe whether the investigators were blinded to group allocation during data collection and/or analysis.

   Not applicable. The entire set of data was analyzed by the respective methodologies presented in our manuscript.

Note: all in vivo studies must report how sample size was determined and whether blinding and randomization were used.

6. Statistical parameters

For all figures and tables that use statistical methods, confirm that the following items are present in relevant figure legends (or in the Methods section if additional space is needed).

n/a	Confirmed
☒	☐ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement (animals, litters, cultures, etc.)
☒	☐ A description of how samples were collected, noting whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ A statement indicating how many times each experiment was replicated
☒	☐ The statistical test(s) used and whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of any assumptions or corrections, such as an adjustment for multiple comparisons
☒	☐ Test values indicating whether an effect is present *Provide confidence intervals or give results of significance tests (e.g. P values) as exact values whenever appropriate and with effect sizes noted.*
☒	☐ A clear description of statistics including central tendency (e.g. median, mean) and variation (e.g. standard deviation, interquartile range)
☒	☐ Clearly defined error bars in all relevant figure captions (with explicit mention of central tendency and variation)

See the web collection on statistics for biologists for further resources and guidance.

Software

Policy information about availability of computer code

7. Software

Describe the software used to analyze the data in this study.	Butler (https://github.com/llevar/butler), R

For manuscripts utilizing custom algorithms or software that are central to the paper but not yet described in the published literature, software must be made available to editors and reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). *Nature Methods* guidance for providing algorithms and software for publication provides further information on this topic.

Materials and reagents

Policy information about availability of materials

8. Materials availability

Indicate whether there are restrictions on availability of unique materials or if these materials are only available for distribution by a third party.	No unique materials were used. All data are available to the community. Algorithms used are distributed as open source.

9. Antibodies

Describe the antibodies used and how they were validated for use in the system under study (i.e. assay and species).	No Antibodies were used.

10. Eukaryotic cell lines

a. State the source of each eukaryotic cell line used.	No eukaryotic cell lines were used.
b. Describe the method of cell line authentication used.	No eukaryotic cell lines were used.
c. Report whether the cell lines were tested for mycoplasma contamination.	No eukaryotic cell lines were used.
d. If any of the cell lines used are listed in the database of commonly misidentified cell lines maintained by ICLAC, provide a scientific rationale for their use.	No commonly misidentified cell lines were used.

Animals and human research participants

Policy information about studies involving animals; when reporting animal research, follow the ARRIVE guidelines

11. Description of research animals

Provide all relevant details on animals and/or animal-derived materials used in the study.	No animals were used.

Policy information about studies involving human research participants

12. Description of human research participants

Describe the covariate-relevant population characteristics of the human research participants.

The PCAWG marker paper presents the population-characteristics of these cancer patients in great detail, see http://www.biorxiv.org/content/biorxiv/early/2017/07/12/162784.full.pdf. In brief, demographically, the cohort included male (55%) and female (45%) donors, with a mean age of 56 years (median 60 years; range 1-90 years). By using population ancestry-differentiated single nucleotide polymorphisms (SNPs), we were able to estimate the population ancestry of each donor. The continental ancestry distribution was heavily weighted towards Europeans (77% of total) followed by East Asians (16%), as expected by large contributions from European, North American, and Australian projects.