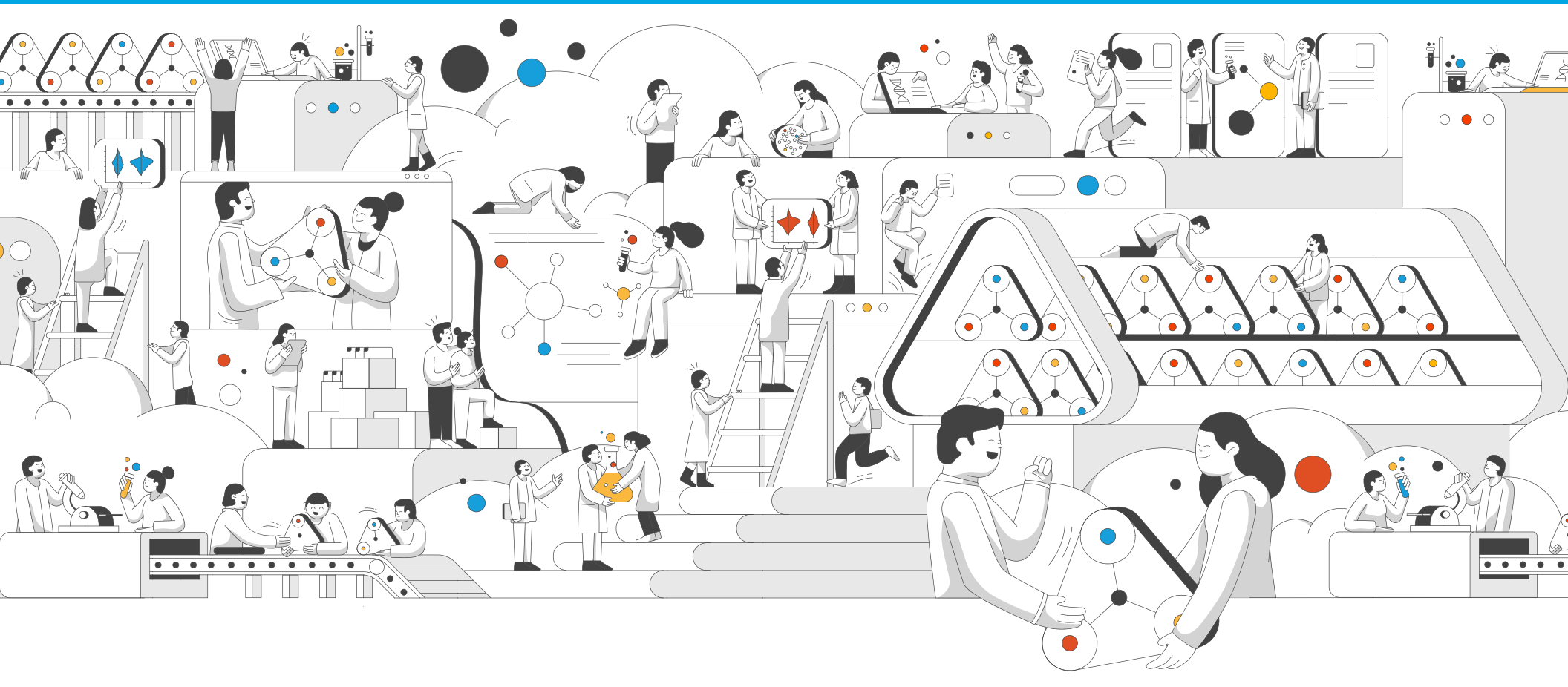
EBOOK

CODE OCEAN

The Three Cornerstones of High-Throughput Collaboration

A Guide to High-Performance Computational Research Teams



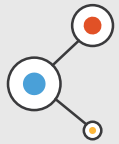
The biotechnology industry has exploded during the last few decades. What started as individual researchers conducting small-scale experiments and analyzing them using spreadsheets has transformed into teams of researchers generating massive amounts of data from relevant human samples using high-throughput technologies and analyzing the data with powerful algorithms.

The ability to successfully navigate this transition requires team leaders to deliberately design modern computational research teams, including the roles, process, and automation, to deliver high-quality research.

Big data, a wealth of sophisticated analysis tools, complex computing environments, and large teams of specialized scientists have created a new bottleneck: **the ability of researchers to continuously collaborate to turn all that data into knowledge and insights.**

This eBook will analyze the three cornerstones that support high-throughput collaborations:

Shareability



Traceability

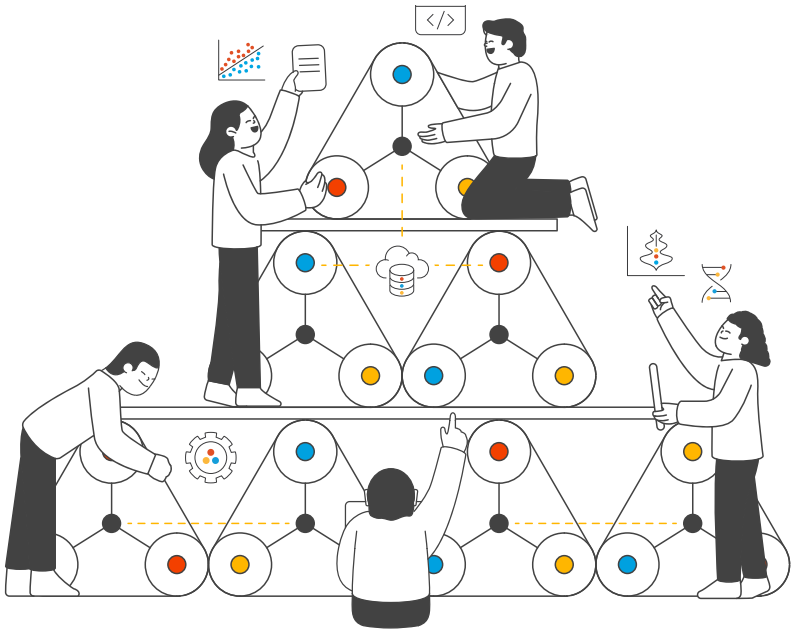
Reproducibility

Continuous collaboration between multidisciplinary and often geographically dispersed teams is challenging. Fundamentally, the challenge is to make sure every researcher working with the data uses not just the same dataset, but also the same analysis packages, the same code, and the same computing environment. This is difficult for several reasons:

- Creating and/or operationalizing the right analysis packages at the right time requires a level of IT expertise and infrastructure competence that many science researchers do not have.
- Maintaining the exact same version of all analysis packages on all computing environments is highly challenging and gets increasingly more difficult as teams grow and span different countries with different languages, laws, and regulatory requirements.
- Security related to sharing data sets is particularly critical given the sensitive nature of healthcare data. However, it is difficult for users to implement and manage security, access control, centralized management, and cost control themselves or to communicate with IT for support.
- For IT, it is difficult to continuously keep track of all the dependencies, configurations, changes, and versions of all the related research artifacts.

These challenges impact reproducibility, slow down progress and innovation, and take up significant IT resources. Because collaborating can be complex and frustratingly slow, research team members often end up developing suboptimal and non-compliant workarounds that create more problems in the long run and/or compromise security.

Solving this issue means providing a new way for research teams to easily share computational research in a highly traceable and reproducible manner without specialized software engineering knowledge. This also frees up IT teams who otherwise need to support them.



Cornerstone 1

Shareability

Effective and continuous collaborations between and among teams require first and foremost that team members can quickly and easily share data, code, computing environment, and results not just within their teams, but also cross-functionally. For example:

- Bioinformaticians need to be able to give computational biologists easy access to their data pipeline so they can build on their results.
- Engineers need to share data and algorithms with bioinformaticians and computational biologists for data preparation and data analysis.
- Computational biologists need to share data, code, and results with researchers in-house and at partner organizations.

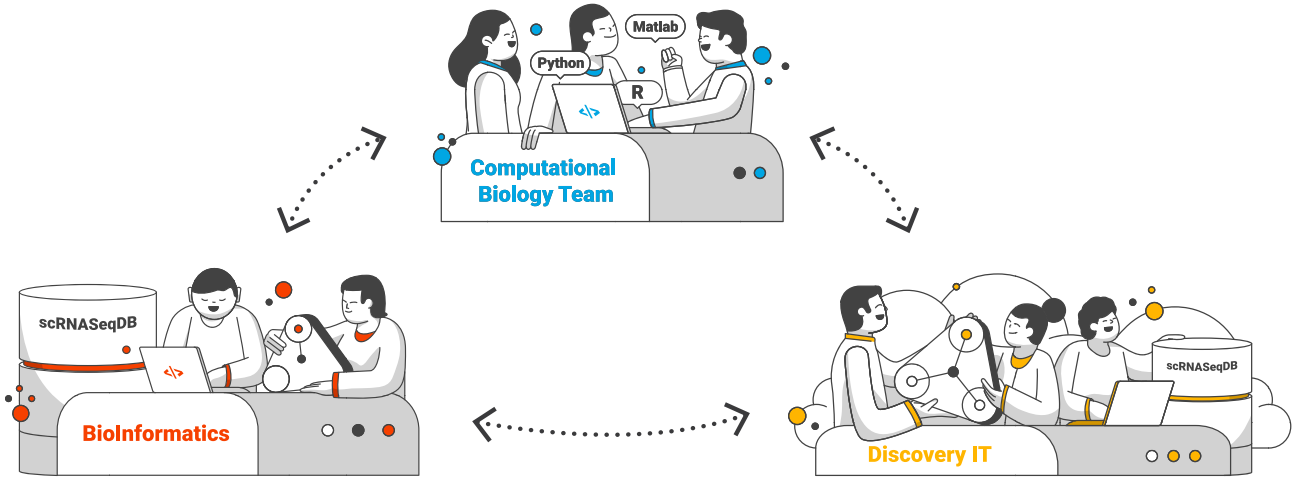
While the ability to share is fundamental to productive collaborations, traditional tools lack the functionality that allows multiple users to work on the same project. Therefore, they fall short of providing a seamless user experience that allows users to readily share their analysis.

As a result, information, such as updates, new data, or additional analyses might not be readily accessible to all collaborators, leading to a host of issues. For example, cumbersome access can delay progress and affect reproducibility. Frustrated users might resort to sharing data in a non-secure and non-compliant manner which not only impacts security, but also leads to version control issues and stale data. Other challenges include:

Complex Processes: Sharing of results currently requires many steps and significant coding experience. This impedes the hand-off between different users, e.g. a bioinformatician's results that serve as input for a computational biologist.

Collaboration Between Coders and Non-Coders: Computational researchers need to work collaboratively with bench scientists who often do not have coding experience.

Sharing with External Partners: Efficient sharing with an external partner is difficult without allowing them into your private space.



Cornerstone 1

Shareability

The Ideal User Experience:

Seamless shareability is critical for running efficient projects as well as avoiding delays and cost overruns. Here are seven characteristics of a desirable user experience:

1. Ability to fork or duplicate entire computational projects with a click and no need to reassemble.
2. Out-of-the-box capabilities to work collaboratively with multiple colleagues, including frictionless sharing and reuse of project work and centralized organizing with intuitive functionality and workflow.
3. Out-of-the-box capability to share apps and functions with a variety of other stakeholders internally and externally such as management, medical, or clinical functions.
4. Out-of-the box user ability to establish multiple roles for different project components (data, code, algorithms).
5. Easy addition and use of standard tools and packages.
6. Availability of a standardized computing environment for all users.
7. Ability to control access to provide a secure and compliant environment.

Key Take-Aways

- Research shareability requires packaging the exact set of related and required data, code, computing environment, and results to ensure successful and on-time completion of computational projects.
- For researchers, an intuitive way to share with colleagues and partners that does not require software engineering expertise or ongoing support by the IT team is important. For IT teams, an intuitive interface that facilitates sharing reduces their workload.
- Easy shareability streamlines analysis collaborations and shortens time to results.

Shareability Matters in Real Life

One person's output is another person's input

CHALLENGE
The results of a bioinformatics pipeline can serve as input for a computational biologist. Sharing the results currently requires many steps and coding experience.

CODE OCEAN PLATFORM
Results are stored as Git assets in a Compute Capsule and can be easily shared with colleagues by sending the relevant URL. Sharing can be managed so only users with the appropriate privileges can access these results. Old and new users can now work collaboratively in a version-controlled environment.

BENEFIT
Seamless and secure access to data, analysis, environment and, importantly, shortened time to results.

Cornerstone 2

Traceability

Scientific progress, new discoveries, and innovations happen when computational researchers collaborate and share analyses. Building on previous results can deliver deeper insights more quickly than starting anew.

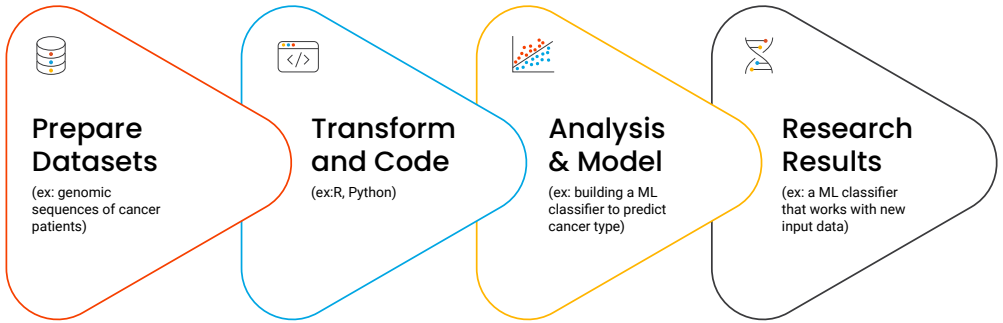
With ever more data and better analysis tools, sharing project data, code, analysis, and results with multiple internal experts and external research partners is increasingly critical to shorten the path to new discoveries and increase the return on research investment. While few would argue against collaborating, in practice, sharing of large data sets, complex analyses, and results is difficult to do continuously and reliably. One of the main challenges is to ensure that all changes to data, results, code, and algorithms are tracked.

The need for high-throughput collaboration requires a high level of coordination and introduces change management requirements. Leaders of computational research teams must address these challenges to ensure that their analysis pipeline can be shared seamlessly and issues with tracking and versioning do not lead to larger problems down the road.

Comprehensive traceability means tracking:

- All code and algorithms
- All data sets
- All analysis results
- Metadata for the analysis
- All tools and packages required to execute the analysis
- The exact compute and environment configurations to execute the analysis
- Security settings for users and access control

A system designed for comprehensive tracking fosters collaboration: the more computational researchers can trust the analysis the faster it can be shared.



Cornerstone 2

Traceability

The Ideal User Experience

The tracking infrastructure needed to support sharing and collaborative work is important, but also difficult to implement and maintain. The two main user groups – researchers and IT professionals – have distinct needs.

Researchers: Tracking Without Knowing How It Happens

Most computational researchers, bioinformaticians, and especially bench scientists are not experienced coders or devops-savvy. They need a tracking infrastructure that supports collaboration but does not impact their work. Ideally, they would know tracking is happening but would never have to worry about how. Tracking requires technology that is best managed by trained IT professionals with devops and coding experience; users without that experience can easily get themselves into trouble using the wrong version of data, code, or environment. Other challenges include inadvertently sharing analyses with the wrong researcher or team, or being unable to get a shared analysis to work.

The ideal workflow is simple and allows a researcher to easily:

- Visualize and understand the versions and history
- Package, commit, and publish any changes to data, code, and environment
- Explore, reuse, and share the right version of the analysis

An intuitive user interface that hides the complexity of tracing allows non-coders to seamlessly perform analyses while critically important tracking tasks happen in the background.

IT: Tracking and Versioning to Keep Teams Collaborating

Transparent tracking needs to happen before, during and after each project and requires application-level interfaces to make it simple for researchers to make changes and share their results. It often falls to IT to put devops processes in place to track assets, maintain them, and/or come to the rescue if computational researchers run into problems. This can lead to bottlenecks and delays of projects because IT is overtasked.

The ideal scenario for IT is a tracking and versioning system that they implement once and maintain as needed. This system should run without intervention by researchers which reduces the risk of problems due to complicated interfaces and inexperienced users. In addition, a good tracking system makes it much easier to support research audits, IT compliance as well as version tracking or reporting.

Improved traceability can address challenges such as:

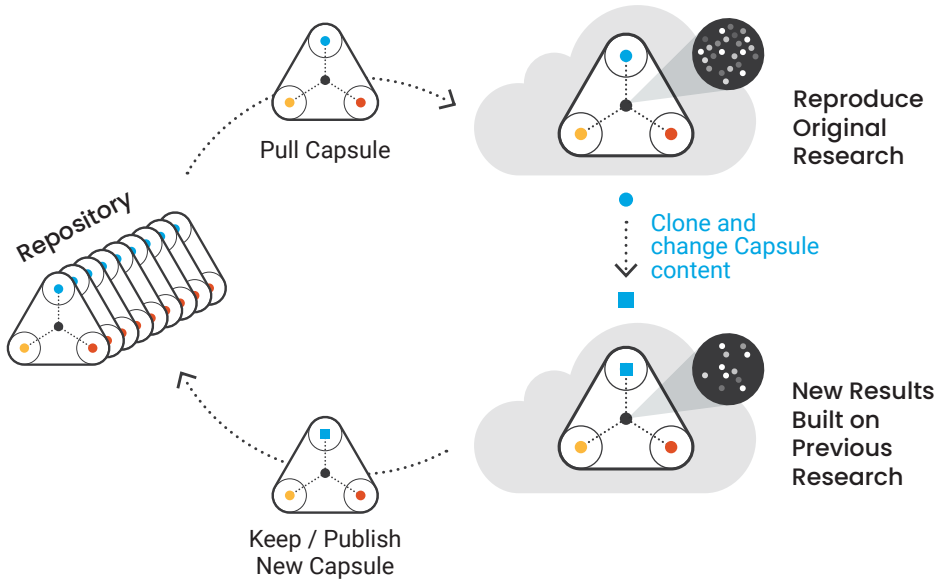
Tracking data graphs: Graphs of important work may make their way into corporate or investor presentations. Once pasted into a slide deck or Word document, keeping track of where the graph came from is challenging. Even slight changes to the graph, such as font size or type, can result in a computational researcher spending hours to find the graph, making the changes and then replacing the original.

Cornerstone 2

Traceability

Key Take-Aways

- Traceability is fundamental to maintaining high-quality assets that are ready for collaborative project work. Tracking lets users know where data files are, who accessed them and what pre-processing was done.
- For researchers, automated tracking is a must as manually managing projects is extremely tedious and error-prone. IT teams need to know that tracking is automated, trusted, and reliable.
- Automated tracking makes it easy to manage multiple researchers using the same data, code, and algorithms removing the time-consuming process of rerunning data, minimizing variability, and avoiding delays.



Traceability Matters in Real Life

Keeping track of versions of large data files

CHALLENGE
Next generation sequencing or proteomics research can generate huge data files. Those data files need to be moved around to various internal locations, uploaded to cloud servers and then moved back to on-premise locations. Keeping track of where the data is and what preprocessing it has undergone is challenging.

CODE OCEAN PLATFORM
Compute Capsules contain all data necessary for an analysis and automatically keep track of any changes such as preprocessing steps. Users do not need to keep logs or implement additional tracking systems; comprehensive tracking is baked into the Compute Capsules.

BENEFIT
Users always know where data files are, who accessed them, and what preprocessing was done without having to actively manage tracking. With automatic tracking all researchers can work and share their results with confidence.

Cornerstone 3

Reproducibility

A survey conducted by Nature[1] finds that "more than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments." Results like these emphasize how wide-spread and serious the problem is.

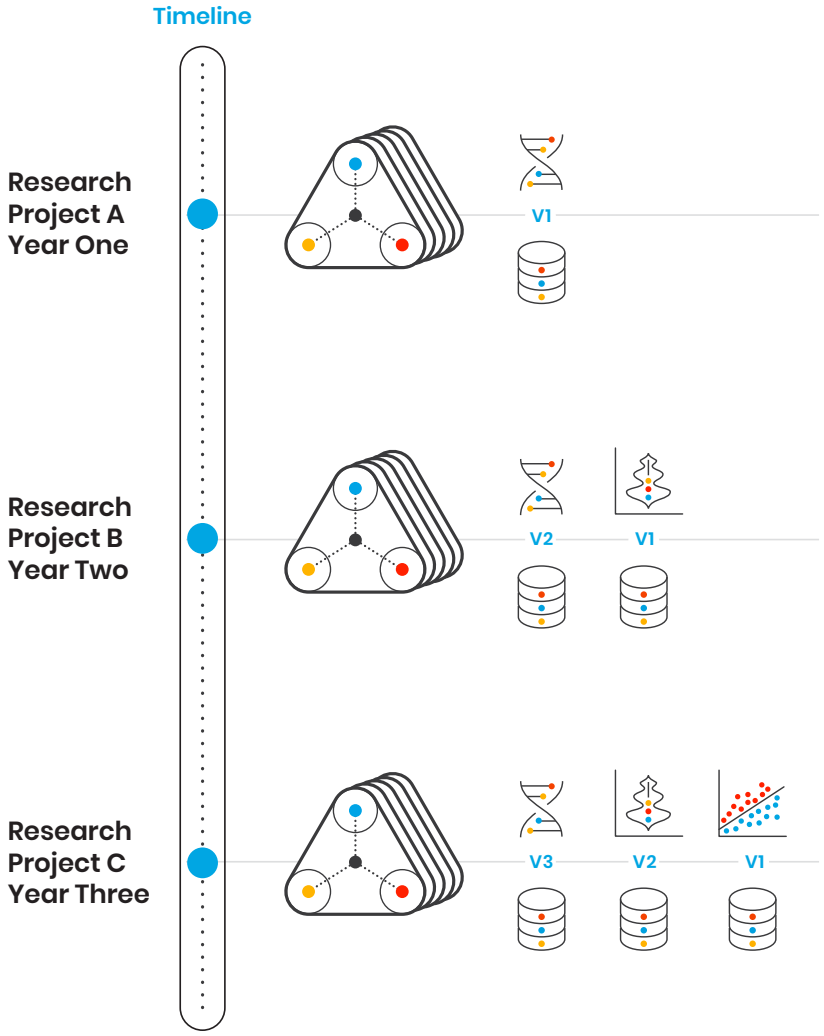
The scientific community is discussing various solutions, from mitigating cognitive biases to improving statistical models and lessening the pressure to "publish or perish." One of the main ways to improve reproducibility is increasing transparency and documenting all analysis steps:

- Data source
- Data pre-processing and cleaning steps
- Analysis pipeline - including software versions, IDEs, visualization tools, and computing languages

More transparency will not only make research more reproducible but will also help computational science teams to collaborate more efficiently: the more readily data, code, and models can be shared, the faster results can be reproduced, and new discoveries can be made based on that foundation of well-documented, transparent prior work.

This level of transparency, however, has historically been difficult to achieve. In fact, there continues to be a large gap between computational researchers' need for reproducibility and IT's ability to understand the intricacies and deliver a practical, robust, and easy to use solution.

This reproducibility gap needs to be closed before high-throughput collaboration can become a reality.



[1] Baker, M. 1,500 scientists lift the lid on reproducibility. Nature 533, 452–454 (2016). https://doi.org/10.1038/533452a

Cornerstone 3

Reproducibility

The Ideal User Experience:

Ideally, computational research teams would have easy access to the exact data, code, and computing environment that were used to create a result, even if that result was generated months or years ago. IT teams would be able to provision and manage guaranteed reproducibility without the burden of unsustainable amounts of cumbersome devops infrastructure and hard-to-track documentation.

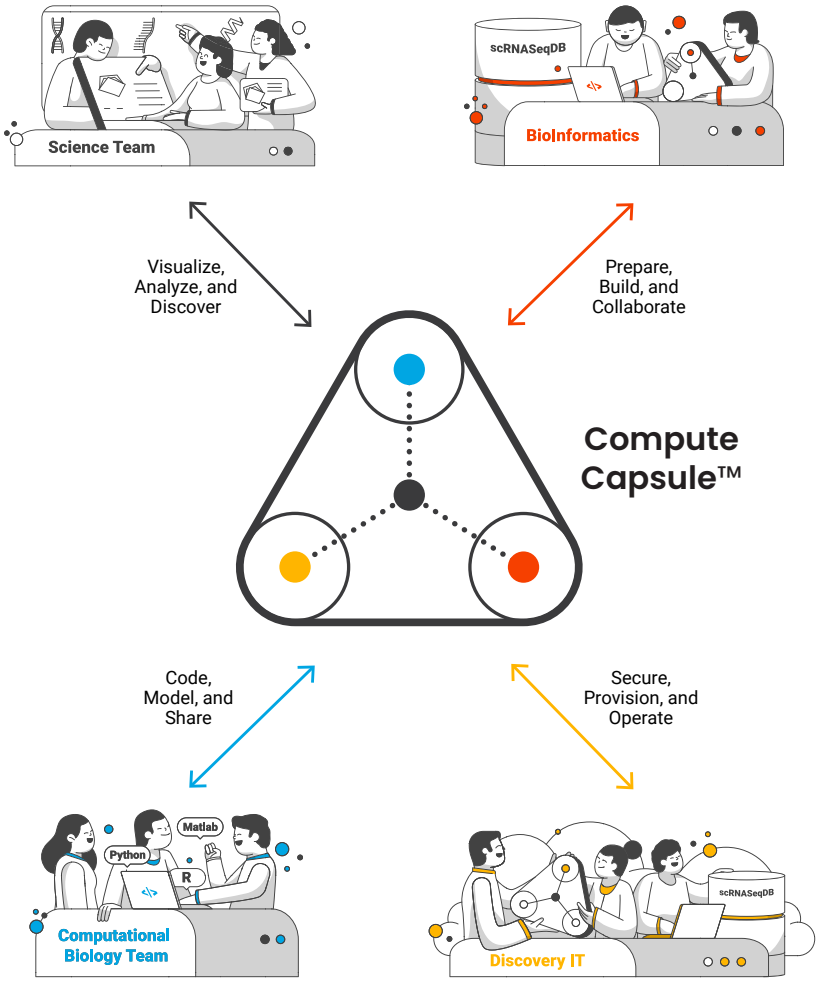
Researchers: Hit RUN and Get the Same Results

The ideal reproducibility experience for a computational researcher is generating the same exact results with the same data and code as any other person running the analysis at any time without the need to debug or make any changes. In short: just hit RUN and the results are the same.

For this to work, each collaborating researcher needs access to:

- All data used for an analysis without changes
- All code and algorithms in exactly the versions that were used in the initial analysis
- The analysis pipeline that was used in the initial analysis
- All results generated in the analysis

> *"The first step of reusing and building on previous work is reproducibility."*
>
> Simon Adar, CEO, Code Ocean



Cornerstone 3

Reproducibility

IT: Ensure Hassle-Free Reproducibility

From an IT perspective, the availability of a platform for packaging all the analysis and the execution environment ensures research can be always be reproduced.

Closing the usability gap between researchers and IT infrastructure and operations optimizes the user experience for both teams.

While open-source notebooks such as Jupyter Notebook address some of these needs, they fall short of delivering complete guaranteed reproducibility and ease of use.

> ***... computational notebooks can be confusing and foster poor coding practices. Case in point, a 2019 study found that just 24% of 863,878 publicly available Jupyter notebooks on GitHub could be successfully re-executed, and only 4% produced the same results.***
>
> J. F. Pimentel et al. in 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) 507–517; IEEE, 2019)

These examples highlight where the lack of reproducibility impacts the work of researchers:

Failed Re-Runs: Running code after a period of months or even years have passed often fails to reproduce the original results. Extensive trouble-shooting and even complete recreation of the analysis pipeline might be necessary.

Researcher or Team Turnover: Replicating an analysis after a team member leaves the company is virtually impossible because the documentation and environment are incomplete. At best, it will take a significant amount of time and effort to recreate the analysis, at worst the analysis can't be replicated, and the project needs to be restarted or dropped.

Replicability Across Analysis Pipelines: Analysis pipelines developed by computational researchers are often specific for one project. This one-time code approach means that no standard analyses are established across the company, jeopardizing reproducibility, making it harder to compare results, and wasting resources re-writing similar code.

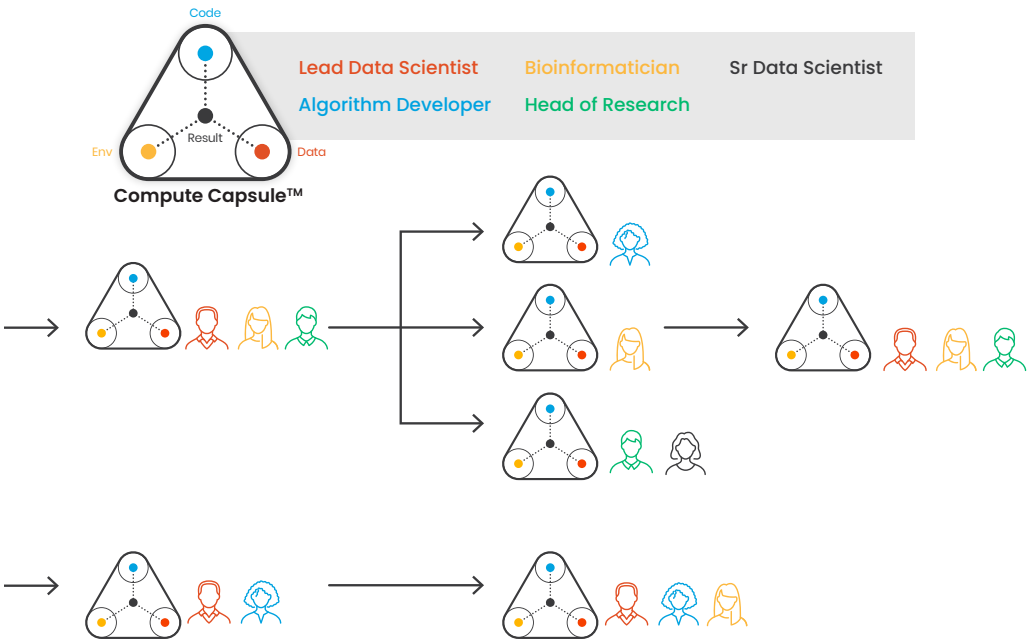
Cornerstone 3

Reproducibility

Key Take-Aways

- To effectively address the reproducibility crisis increased transparency is vital. This includes documenting all steps of an analysis including which code, data and environment was used and how they were modified and applied.
- For researchers, the ideal scenario is a system that allows them to get exactly the same results by simply hitting "rerun" regardless of when and by whom the analysis was originally conducted.
- Reproducibility enables standardized analysis and improves the consistency and comparability of results.

Different roles duplicating and sharing research during a project



Reproducibility Matters in Real Life

Leveraging an Analysis Pipeline as Template

CHALLENGE
Analysis pipelines developed by computational scientists are often specific for one project. This one-time code approach means that no standard analyses are established across company, jeopardizing reproducibility, making it harder to compare results, and wasting resources re-writing similar code.

CODE OCEAN PLATFORM
Once an analysis pipeline has been developed it can be added to a Compute Capsule and shared between researchers. Broadly useful Capsules can be further optimized and shared as templates that make future analysis work faster, easier, and more reproducible.

BENEFIT
By using Compute Capsules to share and develop analyses companies can standardize approaches and improve consistency and comparability. Compute Capsules make it easy to preserve, improve upon, and share important work while accelerating the time to result.

CODE OCEAN

Code Ocean Platform

Code Ocean's platform was built to allow efficient and transparent high-throughput collaboration. If you would like to discuss with one of our team members how the Code Ocean Workbench, Compute Capsules™ and App Panel can improve reproducibility, shareability and traceability of your computational work, please contact us at:

info@codeocean.com
www.codeocean.com

Click here to chat with us and see how Code Ocean can support your collaboration needs

REQUEST A DEMO