Novo Nordisk Foundation NNF10CC1016517 and the Keck Foundation; A.R. received a Lilly Innovation Fellowship Award; B.G.-J. and J. Nogales received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 686585 for the project LIAR, and the Spanish Ministry of Economy and Competitivity through the RobDcode grant (BIO2014-59528-JIN); L.M.B. has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 633962 for project P4SB; R.F. received funding from the US Department of Energy, Offices of Advanced Scientific Computing Research and the Biological and Environmental Research as part of the Scientific Discovery Through Advanced Computing program, grant DE-SC0010429; A.M., C.Z., S.L. and J. Nielsen received funding from The Knut and Alice Wallenberg Foundation, Advanced Computing program, grant #DE-SC0010429; S.K.'s work was in part supported by the German Federal Ministry of Education and Research (de.NBI partner project "ModSim" (FKZ: 031L104B)); E.K. and J.A.H.W. were supported by the German Federal Ministry of Education and Research (project "SysToxChip", FKZ 031A303A); M.K. is supported by the Federal Ministry of Education and Research (BMBF, Germany) within the research network Systems Medicine of the Liver (LiSyM, grant number 031L0054); J.A.P. and G.L.M. acknowledge funding from US National Institutes of Health (T32-LM012416, R01-AT010253, R01-GM108501) and the Wagner Foundation; G.L.M. acknowledges funding from a Grand Challenges Exploration Phase I grant (OPP1211869) from the Bill & Melinda Gates Foundation; H.H. and R.S.M.S. received funding from the Biotechnology and Biological Sciences Research Council MultiMod (BB/N019482/1); H.U.K. and S.Y.L. received funding from the Technology Development Program to Solve Climate Changes on Systems Metabolic Engineering for Biorefineries (grants NRF-2012M1A2A2026556 and NRF-2012M1A2A2026557) from the Ministry of Science and ICT through the National Research Foundation (NRF) of Korea; H.U.K. received funding from the Bio & Medical Technology Development Program of the NRF, the Ministry of Science and ICT (NRF-2018M3A9H3020459); P.B., B.J.S., Z.K., B.O.P., C.L., M.B., N.S., M.H. and A.F. received funding through Novo Nordisk Foundation through the Center for Biosustainability at the Technical University of Denmark (NNF10CC1016517); D.-Y.L. received funding from the Next-Generation BioGreen 21 Program (SSAC, PJ01334605), Rural Development Administration, Republic of Korea; G.F. was supported by the RobustYeast within ERA net project via SystemsX.ch; V.H. received funding from the ETH Domain and Swiss National Science Foundation; M.P. acknowledges Oxford Brookes University; J.C.X. received support via European Research Council (666053) to W.F. Martin; B.E.E. acknowledges funding through the CSIRO-UQ Synthetic Biology Alliance; C.D. is supported by a Washington Research Foundation Distinguished Investigator Award. I.N. received funding from National Institutes of Health (NIH)/National Institute of General Medical Sciences (NIGMS) (grant P20GM125503).

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41587-020-0446-y.

Check for updates

The nf-core framework for community-curated bioinformatics pipelines

To the Editor — The standardization, portability and reproducibility of analysis pipelines are key issues within the bioinformatics community. Most bioinformatics pipelines are designed for use on-premises; as a result, the associated software dependencies and execution logic are likely to be tightly coupled with proprietary computing environments. This can make it difficult or even impossible for others to reproduce the ensuing results, which is a fundamental requirement for the validation of scientific findings. Here, we introduce the nf-core framework as a means for the development of collaborative, peer-reviewed, best-practice analysis pipelines (Fig. 1). All nf-core pipelines are written in Nextflow and so inherit the ability to be executed on most computational infrastructures, as well as having native support for container technologies such as Docker and Singularity. The nf-core community (Supplementary Fig. 1) has developed a suite of tools that automate pipeline creation, testing, deployment and synchronization. Our goal is to provide a framework for high-quality bioinformatics pipelines that can be used across all institutions and research facilities.

Being able to reproduce scientific results is the central tenet of the scientific method. However, moving toward FAIR (findable, accessible, interoperable and reusable) research methods[1] in data-driven science is complex[2,3]. Central repositories, such as bio.tools[4], omictools[5] and the Galaxy toolshed[6], make it possible to find existing pipelines and their associated tools. However, it is still notoriously challenging to develop analysis pipelines that are fully reproducible and interoperable across multiple systems and institutions — primarily because of differences in hardware, operating systems and software versions.

Although the recommended guidelines for some analysis pipelines have become standardized (for example, GATK best practices[7]), the actual implementations are usually developed on a case-by-case basis. As such, there is often little incentive to test, document and implement pipelines in a way that permits their reuse by other researchers. This can hamper sustainable sharing of data and tools, and results in a proliferation of heterogeneous analysis pipelines, making it difficult for newcomers to find what they need to address a specific analysis question.

As the scale of -omics data and their associated analytical tools has grown, the scientific community is increasingly moving toward the use of specialized workflow management systems to build analysis pipelines[8]. They separate the requirements of the underlying compute infrastructure from the analysis and workflow description, introducing a higher degree of portability as compared to custom in-house scripts. One such popular tool is Nextflow[9]. Using Nextflow, software packages can be bundled with analysis pipelines using built-in integration for package managers, such as Conda, and containerization platforms, such as Docker and Singularity. Moreover, support for most common high-performance-computing batch schedulers and cloud providers allows simple deployment of analysis pipelines on almost any infrastructure. The opportunity to run pipelines locally during initial development and then to proceed seamlessly to large-scale computational resources in high-performance-computing or cloud settings provides users and developers with great flexibility.

The nf-core community project collects a curated set of best-practice analysis pipelines built using Nextflow. Similar projects include the 'awesome-pipelines' repository, which provides an extensive list of pipelines developed by the Nextflow community (https://github.com/pditommaso/awesome-pipeline), although these pipelines are variable in terms of development status and design. High-level approaches to facilitate the creation of end-to-end analysis pipelines are also available: Flowcraft (https://github.com/assemblerflow/flowcraft) and Pipeliner[10] are

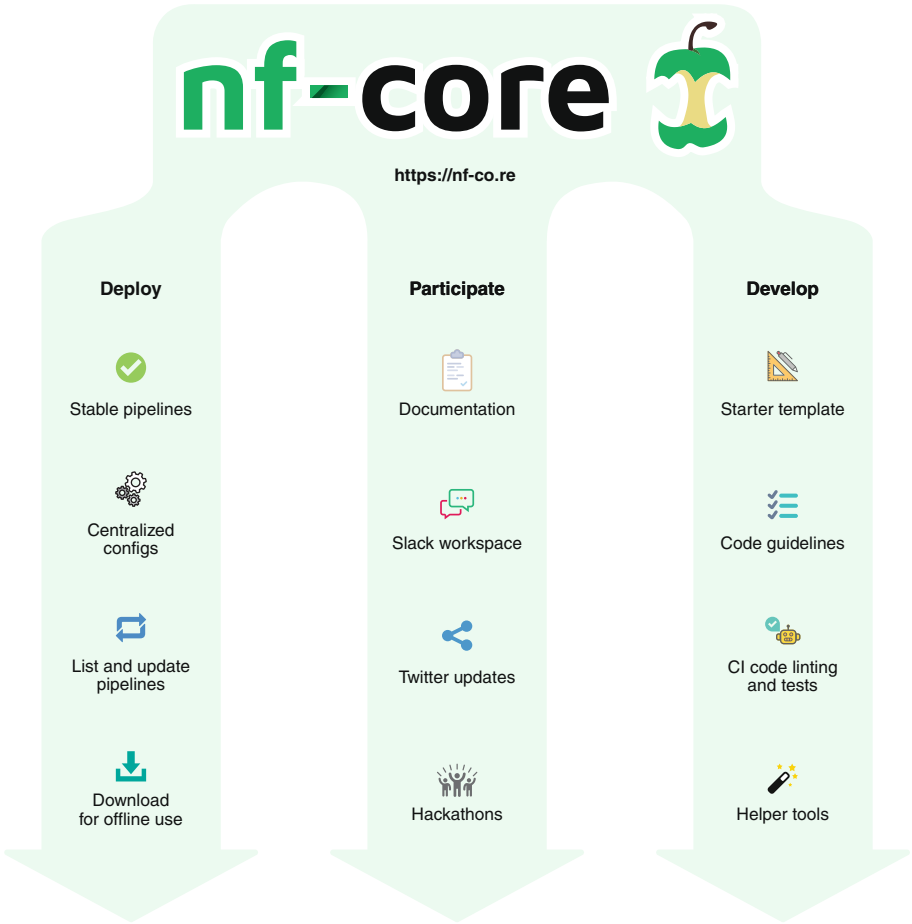

Fig. 1 | Main concepts of nf-core. Best-practice pipelines are available to be deployed on virtually any computational infrastructure. Community-built tools help pipeline developers to create new pipelines and adhere to nf-core guidelines. Slack, Twitter and events such as hackathons allow both users and developers to actively participate in the nf-core community. CI, continuous integration.

meta-tools to dynamically build Nextflow pipelines from modular template blocks. Snakemake Workflows (https://github.com/snakemake-workflows) is an early-stage community project attempting to collect best-practice analysis pipelines based on the Snakemake[11] workflow manager (see "Comparison to other projects" in the Supplementary Note). The Galaxy Toolshed[6] for Galaxy[12] and the ENCODE project pipelines[13] collect pipelines for their respective platforms. The nf-core project, however, is unique in its use of strict development best-practice guidelines, pipeline templates, testing and automation tools, and comprehensive documentation for both developers and end users (see "Guidelines" and "Tools" in the Supplementary Note, and Supplementary Fig. 2). By leveraging the functionality offered by Nextflow and mandating best practices through the use of community-built tools, nf-core is able to provide cutting-edge analysis pipelines that are truly portable and reproducible.

The primary portal to the nf-core community is its website (https://nf-co.re), which lists available analysis pipelines, user- and developer-centric documentation, and tutorials, as well as usage and contributor statistics. All code is hosted on GitHub under the nf-core organization (https://github.com/nf-core/) and released under the highly permissive MIT license, encouraging downstream modification and reuse.

At the time of writing, nf-core encompasses a total of 35 analysis pipelines (see "Pipelines" in the Supplementary Note). Most nf-core pipelines perform extensive primary analyses that provide scientists with a good starting point to focus on the interrogation of results within the biological context of their experiment. Although being strongly represented in the analysis of genomics data, the nf-core framework is also being used to develop pipelines for the interpretation of proteomics (nf-core/mhcquant and nf-core/ddamsproteomics) and image analysis experiments (nf-core/imcyto). This highlights the flexibility of the framework and its applicability to other areas within the broader field of scientific data analysis.

Pipelines within nf-core must adhere to strict guidelines (Supplementary Table 1), many of which are formulated according to published scientific-computing best practices[14–16]. These guidelines are highlighted on the website and described in more detail in the Supplementary Note under "Guidelines." All pipelines are required to provide high-quality documentation: pipeline usage documentation with examples and also a description of the generated results files. Test datasets must be provided in order to run automated continuous-integration tests whenever there is a change to a pipeline, ensuring that developers can guarantee a working pipeline at all times. Where applicable, common usage parameters are imposed across nf-core pipelines so that, once a researcher is able to successfully run a single nf-core analysis pipeline, others will work on the same system with minimal command-line alterations. On every pipeline release, a unique version tag is assigned that provides a static link between the pipeline's implementation and its associated software dependencies (see "Reproducibility" in the Supplementary Note). Additionally, automated linking mechanisms between GitHub and Zenodo[17] allow each pipeline release to be citable via a unique DOI. This provides users with the reassurance that the results will be fully reproducible as well as citable.

To help developers get started with new pipelines, nf-core provides a standardized pipeline template that adheres to all nf-core guidelines (see "Guidelines" and "Tools" in the Supplementary Note). The use of this template lowers the learning curve when getting started with Nextflow and helps to homogenize the coding style of all nf-core pipelines. Moreover, early feedback from the community and the successive reviewing process allow developers to improve their analysis pipelines and ensure that they adhere to community standards.

The nf-core pipeline template changes over time, and new best-practice insights need to be integrated into existing pipelines. As a result, we provide an automated synchronization mechanism that distributes the relevant changes to existing pipelines at every new release of the template (Supplementary Fig. 3). The pipeline maintainers can review the

suggested changes and merge them into the development version of the source code, updating the pipeline with minimal manual effort.

The driving force behind nf-core is its community (see "Community" in the Supplementary Note). Through open discussion and collaboration among the community, it is possible to leverage the knowledge of experts across the world for the development of domain-specific pipelines and the implementation of current best-practice analysis methods. This collaborative process bypasses the traditional barriers that can exist between research groups, resulting in high-quality pipelines that anyone can use. Pipelines are reviewed and used by contributors from multiple institutions, ensuring that they are tested on a range of computational infrastructures with data from a variety of sources. Code is always reviewed by more than one person, as is best practice for scientific computing[16]. The nf-core project ties in well with other communities, notably Nextflow[9], Bioconda[18] and conda-forge (https://conda-forge.org/). Wherever possible, we encourage users and developers to contribute to these projects, such as by packaging missing software in the Bioconda or conda-forge repositories, benefitting the wider bioinformatics community.

As the usage of workflow management tools spreads, an increasing number of tertiary tools are tying into the ecosystem. The nf-core analysis pipelines are at the forefront of this, collaborating with initiatives such as bio.tools[4] and the GA4GH-compliant Dockstore[19], as well as having plans to work together with the Biocontainers[20] project to further simplify software packaging.

We are in the process of building an interactive command-line and graphical user interface to further simplify the process of launching nf-core pipelines. We are extending our testing infrastructure to perform automated benchmarking using full-size test datasets and plan to develop tools that provide more accurate cloud computing price estimates. An important shift in Nextflow pipeline best practices will be realized with the upcoming release of the Nextflow DSLv2 modular language, which will be implemented across nf-core pipelines to improve code readability and reuse. Looking ahead, we hope to welcome more contributors and pipelines to the nf-core community to build on the solid foundation that has already been established.

Data availability

No datasets were generated or analyzed during this study.

Code availability

The source code for the nf-core framework and all nf-core pipelines is publicly available at https://github.com/nf-core/. Where applicable, Zenodo DOIs are available on the respective pipeline repositories. ❐

Philip A. Ewels[1,9], Alexander Peltzer[2,9], Sven Fillinger[2], Harshil Patel[3], Johannes Alneberg[4], Andreas Wilm[5], Maxime Ulysse Garcia[6], Paolo Di Tommaso[7,8] and Sven Nahnsen[2]✉

[1]Science for Life Laboratory (SciLifeLab), Department of Biochemistry and Biophysics, Stockholm University, Stockholm, Sweden. [2]Quantitative Biology Center (QBiC), University of Tübingen, Tübingen, Germany. [3]Bioinformatics and Biostatistics, The Francis Crick Institute, London, UK. [4]Science for Life Laboratory (SciLifeLab), School of Engineering Sciences in Chemistry, Biotechnology and Health, Department of Gene Technology, Royal Institute of Technology, Stockholm, Sweden. [5]Computational & Systems Biology, Genome Institute of Singapore, Singapore, Singapore. [6]Department of Oncology-Pathology, Karolinska Institutet, Stockholm, Sweden. [7]Centre for Genomic Regulation (CRG), The Barcelona Institute for Science and Technology, Barcelona, Spain. [8]Universitat Pompeu Fabra (UPF), Barcelona, Spain. [9]These authors contributed equally: Philip A. Ewels, Alexander Peltzer.

✉*e-mail: sven.nahnsen@qbic.uni-tuebingen.de*

Published online: 13 February 2020
https://doi.org/10.1038/s41587-020-0439-x

References

1. Sansone, S.-A. et al. *Nat. Biotechnol.* **37**, 358–367 (2019).
2. Perkel, J. M. *Nature* **560**, 513–515 (2018).
3. Baker, M. *Nature* **533**, 452–454 (2016).
4. Ison, J. et al. *Nucleic Acids Res.* **44**, D38–D47 (2016).
5. Henry, V.J., Bandrowski, A.E., Pepin, A.-S., Gonzalez, B.J. & Desfeux, A. *Database (Oxford)* https://doi.org/10.1093/database/bau069 (2014).
6. Blankenberg, D. et al. *Genome Biol.* **15**, 403 (2014).
7. Van der Auwera, G. A. et al. *Curr. Protoc. Bioinforma.* **43**, 11.10.1–11.10.33 (2013).
8. Leipzig, J. *Brief. Bioinform.* **18**, 530–536 (2017).
9. Di Tommaso, P. et al. *Nat. Biotechnol.* **35**, 316–319 (2017).
10. Federico, A. et al. *Front. Genet.* **10**, 614 (2019).
11. Köster, J. & Rahmann, S. in *German Conference on Bioinformatics 2012* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012).
12. Afgan, E. et al. *Nucleic Acids Res.* **46**, W537–W544 (2018).
13. Sloan, C. A. et al. *Nucleic Acids Res.* **44**, D726–D732 (2016).
14. Grüning, B. et al. *Cell Syst.* **6**, 631–635 (2018).
15. Möller, S. et al. *Data Sci. Eng.* **2**, 232–244 (2017).
16. Wilson, G. et al. *PLoS Biol.* **12**, e1001745 (2014).
17. Potter, M. & Smith, T. *Zenodo* https://doi.org/10.5281/zenodo.45042 (2015).
18. Grüning, B. et al. *Nat. Methods* **15**, 475–476 (2018).
19. O'Connor, B. D. et al. *F1000 Res.* **6**, 52 (2017).
20. da Veiga Leprevost, F. et al. *Bioinformatics* **33**, 2580–2582 (2017).

Acknowledgements

The authors would also like to thank the following people for their contributions: F. Bonath, O. Contreras-López, S. Haglund, R. Hammarén, A. Jemt, R. A. Olsen, S. Paneerselvam, M. Proks, J. Wan, C. Wang, J. Westholm and D. Yuen (Science for Life Laboratory, Stockholm, Sweden); C. C. Shih (A*STAR Genome Institute of Singapore); M. Hoeppner (Institut für Klinische Molekularbiologie, Kiel, Germany); V. Malladi (University of Texas Southwestern Medical Center, Dallas); A. Duncan (Ontario Institute for Cancer Research); H. Gourlé (Swedish University of Agricultural Sciences, Uppsala); G. Gabernet, S. Heumos, T. Koch, C. Mohr and D. Straub (Quantitative Biology Centre, Tübingen, Germany); G. Kelly (Francis Crick Institute London); Q. Zhao (Sun Yat-sen University Cancer Center, Guangzhou, China); and E. Floden (Comparative Bioinformatics Laboratory Centre for Genomic Regulation (CRG), Barcelona, Spain). A.P., S.F. and S.N. acknowledge funding from the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG; core facilities initiative, KO-2313/6-1 and KO- 2313-2, Institutional Strategy of the University of Tübingen, ZUK 63). A.P. and S.N. acknowledge funding by the Sonderforschungsbereich SFB/TR 209 "Liver cancer" of the DFG. S.N. acknowledges funding from the DFG Project ID 398967434 – TRR 261 and the DFG im Rahmen der Exzellenzstrategie des Bundes und der Länder EXC 2180 – 390900677 and EXC 2124 – 390838134. P.D. acknowledges the support of the Spanish Ministry of Economy, Industry and Competitiveness (MEIC) to the EMBL partnership, the Centro de Excelencia Severo Ochoa and the CERCA Programme / Generalitat de Catalunya. P.D.'s work is supported by the European Union's Horizon 2020 research and innovation program under grant agreement 815668 (BovReg), the Spanish Ministry of Economy, Industry and Competitiveness (BFU2017-88264-P) and the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya (2017 SGR 447).

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41587-020-0439-x.