Published in final edited form as:
J Stat Data Sci Educ. 2023 ; 31(1): 57–65. doi:10.1080/26939169.2022.2118646.

Open-source Tools for Training Resources - OTTR

Candace Savonen[*,1,2,†], **Carrie Wright**[1,2,†], **Ava M. Hoffman**[1,2,†], **John Muschelli**[1], **Katherine Cox**[1], **Frederick J. Tan**[3], **Jeffrey T. Leek**[1,2,†]

[1]Johns Hopkins Bloomberg School of Public Health, Baltimore, MD

[2]Fred Hutchinson Cancer Center, Seattle, WA

[3]Carnegie Institution, Baltimore, MD

Abstract

Data science and informatics tools are developing at a blistering rate, but their users often lack the educational background or resources to efficiently apply the methods to their research. Training resources and vignettes that accompany these tools often deprecate because their maintenance is not prioritized by funding, giving teams little time to devote to such endeavors. Our group has developed Open-source Tools for Training Resources (OTTR) to offer greater efficiency and flexibility for creating and maintaining these training resources. OTTR empowers creators to customize their work and allows for a simple workflow to publish using multiple platforms. OTTR allows content creators to publish training material to multiple massive online learner communities using familiar rendering mechanics. OTTR allows the incorporation of pedagogical practices like formative and summative assessments in the form of multiple choice questions and fill in the blank problems that are automatically graded. No local installation of any software is required to begin creating content with OTTR. Thus far, 15 training courses have been created with OTTR repository template. By using the OTTR system, the maintenance workload for updating these courses across platforms has been drastically reduced. For more information about OTTR and how to get started, go to ottrproject.org.

Keywords

education; informatics; tools; open-source; data science

1 Introduction

Informatics, data science, and machine learning tools are developing at a dramatic rate. Driven by the rapidly changing technologies that underlie informatics and data science, such

[*]Corresponding author: csavonen@fredhutch.org.
[†]Previously the author's affiliation was Johns Hopkins Bloomberg School of Public Health, but over the course of the development of this manuscript, their affiliation has changed to Fred Hutchinson Cancer Center

Declaration of interest statement
The authors declare no competing interests. Individual courses we have created on Coursera and Leanpub do generate course revenue, but we do not obtain revenue for any courses other individuals create using OTTR.

SUPPLEMENTARY MATERIAL
ottrpal is available on CRAN

as genomic sequencers and imaging technologies, tool developers are constantly working toward new software and updates to handle these data. However, their users often lack the educational background or resources to efficiently apply these quickly evolving methods to their research (Attwood et al. 2017, Wren 2016). Training resources and documentation for these tools are too often non-existent or left to deprecate (Mangul et al. 2019). Those with the deepest and most up-to-date knowledge of the tools, such as developers and power users, generally have limited time or resources to devote to training efforts (Kaiser 2016). Tutorials and courses are valuable additions for making software tools more approachable to users, but the creation of these courses involves time spent on tedious, non-content related tasks that most developers do not have the time or background knowledge to engage in. Our goal is to create technology that addresses these challenges by reducing time spent on tedious, non-content related tasks so instructors and developers can focus on training content.

Unfortunately, a large portion of time spent on training material creation is not related to content and does not require content-specific knowledge. Time is instead spent determining where and how to publish and maintain the material as opposed to what the material should contain. These endeavors are especially time consuming for individuals if they have no training in creating online educational content. These tasks over the life of a course quickly add up in the amount of time and mental energy they exact on a course creator – perhaps the mental energy toll being the larger cost. Overall, this creates a frustrating and inefficient dynamic for both developers and their end users.

Training communities such as The Carpentries and Galaxy Training Network have led the way in creating community resources to improve the development and maintainability of informatics and data science education (Wilson 2016, Batut et al. 2018). These organizations have their own sets of tools, guidelines and templates for individuals who would like to contribute content to their community platform (See Supplementary Table 1). We have created similar tools and templates that allow for large-scale, flexible, and customize-able publishing in a simplified way as they are…

- Compatible with massive open online course platforms like Coursera and Leanpub which have millions of users (Coursera 2021, Leanpub 2021)
- Designed to encourage customization of branding and publishing platform options.
- Built for easy maintenance to encourage the upkeep of the material for a longer period of time.

We have developed the Open-source Tools for Training Resources (OTTR) to offer the efficiency of templated setups that The Carpentries and Galaxy Training Network have, but with a focus on creating customizable Massive Open Online Courses (MOOCs). OTTR is a system of tools, templates, and guides that supports course creation in addition to publication to multiple massive online learner communities that already exist on Coursera and Leanpub. OTTR’s compatibility with Coursera and Leanpub provides a method for incorporating pedagogical practices like formative and summative assessments in the form of multiple choice questions and fill in the blank problems that are automatically graded. MOOCs like Coursera and Leanpub help democratize education by allowing learners to acquire certification for completing courses in a way that is well-recognized and can help learners improve their employment opportunities (Rivas et al. 2020). OTTR also enables customization of style and course accessories so that courses can be tailored for their context, including both the course's specific audience and content creator's institution. Importantly, our tools offer developers the opportunity to write content once and publish in multiple ways if they choose. Our tools and template also allow users flexibility in how they would like their content published. Users can pick between publishing methods and strategies, including what platforms to publish on and if they would like to include quizzes or not. Rather than being a single resource, OTTR is a collection of resources which users can pick and choose from based on their needs.

Anyone can create material through OTTR at any time without prior approval or additional accounts; the only requirement is a free GitHub account. We believe OTTR builds upon existing models of content creation to strengthen the dynamic between developers and users, while also improving the democratization of informatics and data science-related education (Kross et al. 2020).

1.1 OTTR Design Philosophy

- **Maintainable** - After creating training content, informatics tool developers and content creators often move on to other projects or have other priorities take the forefront. We aim to make the workload for future course updates and maintenance as light as possible to deter deprecation of the training resources. Building as much as possible programmatically from plain text (e.g. R Markdown) using Docker images produces the same content every time while ensuring proper environment and version control (Merkel 2014). Easy maintenance also allows developers and content creators to focus on the cutting edge of their fields, rather than time-consuming upkeep of training materials across platforms. The pull request model of GitHub also allows contributions from the community, which could reduce the burden on the original tool developers and increase the robustness of the material. OTTR's automatic checks can help polish any incoming contributions, helping alleviate the cognitive load.
- **Adaptable** - Training resources might be needed in many different contexts. We aim to be prescriptive but flexible. Permissive licensing like CC-BY (a Creative Commons Attribution License) allows content creators to quickly mix and match snippets of code or subsets of slides and customize for a specific audience. All automated tools are customize-able from a simple central configuration file that a user can set on and off tools depending on their course's needs. From this file, users can turn off or on the renderings and checks for publishing to one platform or the other if they do not want to publish to all three platforms. This means the user can avoid the risk of being entirely committed to the file formats required by a particular platform as they are creating content. Furthermore, course chapters that are useful in multiple courses, can be borrowed from one origin minimizing redundant maintenance.

- **Open source and free** - Funding is not always available for developing or maintaining training material; we do not want that to be a barrier to use these tools. A lack of funding for developing materials also propagates existing financial inequities among under-served institutions and faculty (Hemming et al. 2019). Ensuring tools like OTTR do not require the purchase of any software can help alleviate these challenges. Using GitHub's free-tier hosting and workflow means its inherently open source and free.
- **Templates for momentum** - Creating material from a "blank slate" is challenging, and decision fatigue can be a hindrance to authoring new training resources. We aim to give course creators sensible defaults and templates so these creators can begin writing content as soon as possible. But we also provide functionality and guidance for users to customize their material as they need, though use of custom style sets and Docker images.

2 Methods

With OTTR, we have devised a system that allows content creators to generally write once, but publish in three places. Bookdown is an R package and system to create content using RMarkdown, which is a text-based format where you can embed code, plots, math equations, references, and a number of other features of word processing. This RMarkdown content can be rendered as a PDF, docx, ePub, or HTML page, so this fits with a number of readers, namely e-readers and website hosting. A Bookdown website is the default OTTR platform and is hosted with GitHub pages upon set up (See the Supplementary Figure 1 for a summary of the set up process). Leanpub and Coursera allow the distribution of this content, either as a book (Leanpub) or a course (Leanpub and Coursera). Publishing with Bookdown, Leanpub, and Coursera not only allows the course content to reach a wider audience but also allows learners to pick the learning platform that best suits their needs or preferences (Xie 2016, Leanpub 2021, Coursera 2021). The Bookdown-based website is similar to a course website that organizes content in a logical way via chapters or modules. Some content creators might find Bookdown to be the most approachable of the three publishing platforms because it only requires one step for user access: navigating to the appropriate URL. Because the Bookdown-rendered website does not require any login or user account, it cannot provide any certificate of completion. In contrast, Leanpub and Coursera platforms offer quizzes and certificates of completion. This allows user audiences to better assess and keep a record of their learning. Leanpub allows content authors to set their own price for end users (even free) while Coursera requires a paid subscription (See Figure 1).

Publishing in all three platforms has many benefits but can quickly become a maintenance burden. OTTR uses automation and focuses on features that can be well-supported across MOOC platforms to minimize these maintenance pains. It also provides built-in flexibility for creators to change or add publishing platforms at any time during the content creation process, minimizing the risk associated with "committing" to a specific platform.

2.1 The Writing Process

The center of the OTTR approach is writing an R Markdown document (RStudio 2022). R Markdown extends the Markdown specification with some special capabilities based in R. Generally speaking, R Markdown allows the author to combine text, images, and code in one plain text document prior to rendering. This allows for flexibility of content; courses are written in traditional Markdown language but can also render example code not only for R but has extension capabilities for other languages (like Python). For an example of what the steps for adding a new chapter looks like see see Supplementary Figure 2.

Prerequisite knowledge for OTTR usage:

- Familiarity with Markdown.
- Familiarity with GitHub fundamentals, or willingness to learn is required.
- Very basic R knowledge is helpful but not required (local R installation is not needed).

OTTR aims to be approachable for users without much experience in these areas, while also enabling more advanced users freedom to customize settings to suit their needs. Depending on the comfort level of individuals with these prerequisites, OTTR has two options for writing methods. The “Entry Level” writing method allows a user to write courses through the GitHub website interface and is recommended for individuals who are not familiar or not interested in using GitHub on a deeper level. The “Advanced” writing method is recommended for users who are already familiar with GitHub. However, creators are not locked in to a particular path and can pivot to the Advanced method at any time as their GitHub skills grow (See Figure 2).

2.2 Rendering and Previewing

Rendering (and re-rendering) is automated by GitHub Actions, which is a service that can be used to automatically run a set of directions or scripts whenever a change is made, meaning that course creators can create content with their preferred text editor. This eliminates the need for local installation of any additional software. Focusing on just the R Markdown files (and not the resulting rendered files) also prevents confusion that may be introduced when tracking binary files from multiple authors’ file changes. Content can be edited directly from the GitHub web editor and pull requests created from the GitHub website.

Imagine you run a restaurant and you want to add a new item to the menu. You would first taste test a few ideas in the kitchen before printing all new menus and going straight to customers’ plates. For software developers, this testing approach is accomplished with pull requests which OTTR functionality utilizes. In this model, proposed changes are saved to GitHub separately from the main live course documents. This allows for review by the original author and others of these proposed changes before they are actually added to the live course. GitHub and its pull request model thus encourage transparency and collaborative review which can greatly enhance the development process (Dabbish et al. 2012, 2013). In Figure 3, we compare the benefits of the pull request model to that of direct changes (not using pull requests). By utilizing this process, OTTR encourages transparency and enhances the polishing abilities of authors before changes are made live in a course.

When content is pushed to an open pull request on an OTTR-templated GitHub repository, a series of automatic GitHub actions are triggered. These actions automatically render the content as HTML using Bookdown on a GitHub Pages website (See Figure 4) (Xie 2016). This rendering includes an option to include other outputs, such as .pdf and .docx files which allow easy sharing with collaborators. Docx files can be uploaded to Google Drive for easy collection of feedback from collaborators. A preview of the latest content and changes are automatically provided in a GitHub comment (See Figure 4).

2.3 Publishing

The automatically rendered Bookdown version of the course is published using GitHub Pages. Optionally, a course creator can connect individual chapters on Leanpub and Coursera following the OTTR Guide. A custom domain can host a published GitHub Pages, so the content creator may choose where to host this material. Once this set up is done, updates made within a chapter are automatically available in Coursera and Leanpub as soon as the changes are made live on the Bookdown course (See Figure 4).

3 OTTR Components

As our team developed our own courses, we looked for opportunities in the process for minimizing the workload of ourselves and other course creators. For each time consuming process identified, we attempt to create a technical solution such as: an automated tool, a template, or a guide to ease the process.

3.1 Automated tools

Not everything can or should be automated, but the more menial the task, the more that automation is not only possible but is preferred to manually performing the task. Manually performing menial tasks not only is an inefficient use of a course creator’s time, but automated processes are generally more thorough at completing and checking certain tasks. For example, HTML pages are automatically rendered by GitHub Actions. This saves the course creator’s time and mental energy but also avoids accidentally missing files to be rendered, and prevents conflicts when the same content has been edited in multiple ways.

Previews and checks of new content happens automatically as changes are proposed in the form of an open pull request. A pull request is a way to highlight prepared changes to the content. This allows others to review and comment on potential changes before they are incorporated into the live content. The OTTR workflow is centered around the pull request to encourage collaborative review. To this end, the automatic checks and previews occur when pull requests are opened and changes are added; the results of these checks are posted for OTTR authors to easily see and download on the pull request page. The outcomes of checks of spelling, quiz formatting, Docker image building, and working URLs are all reported on pull requests (See Figure 5). All these automated tools, checks, and renderings can be turned on or off from a central configuration file.

3.1.1 Reproducibility using Docker—Docker is a containerization tool which allows content creators to use the same computing environment regardless of their local operating system, essentially freezing the environment to complete rendering tasks so that it does not break with future renderings, especially of the same material. Docker is advantageous to use in the content creation process because it is lightweight (i.e., minimal storage needs) and ensures rendering consistency across collaborators on a set of content. In general, OTTR provides and uses a Docker image to render course material automatically using GitHub Actions. Users do not have to interact with Docker at all to write courses with OTTR, it is something used behind the scenes. However, for some advanced use cases, users may find it advantageous to adapt the OTTR Docker image for their own course uses. This Docker image can also be used locally as an image to build upon for other needs, but local Docker installation or knowledge is not a requirement for using OTTR. Along with Git and GitHub, Docker is a key component of reproducible research (Nüst et al. 2020). Containerization avoids mismatches in R version, package behavior, or hierarchical dependency (otherwise known as “dependency hell”). To maximize reproducibility for creators, we have implemented a container specifically for OTTR, based on the Rocker family of Docker containers (Boettiger & Eddelbuettel 2017). Because Docker might be less familiar to creators compared with GitHub or Markdown, our guide also contains further description on how to customize and adapt this Docker image depending on the particular needs of a course.

3.1.2 *ottrpal* Companion Package—The *ottrpal* R package, a companion package we developed for the OTTR template, assists with the publishing process. Without *ottrpal*’s functions, maintaining course material across three different platforms would be prohibitively tedious and error prone. One of the functions *include_slide()* helps users include images from Google slide decks more seamlessly. Using Google Slides for image storage allows for *ottrpal* to update images seamlessly. Users specify alternative text and other code chunk options, but upon re-rendering the images are all automatically updated from the linked Google Slide deck. Thus, the author does not need to copy and paste updated slides, there is one source of truth for embedded images (in the Google Slide deck), and the content is referenced where it came from. This *ottrpal*/Google Slide system for image storage is still ultimately version controlled; the images retrieved by *ottrpal* are stored in GitHub upon re-rendering. Google Slide is approachable for users in that it is free, user-friendly and built for easy collaboration amongst authors. It also makes for easy creation of presentations related to the content. Google Slide speaker notes also allow the linking of metadata about the slide that can seen as additional documentation for content.

After an initial set up on Leanpub and Coursera, no manual updating is needed for typical updates through the use of embed windows (iframes) on these platforms that link to the main Bookdown content. This means Coursera and Leanpub are instantly refreshed upon course updates to the course on GitHub. The *ottrpal* package creates these embed window versions for Coursera and Leanpub automatically. This simplifies maintenance across platforms so chapter updates do not require extra manual steps (Figure 6).

The *ottrpal* package also facilitates format conversion and quiz creation for uploading to Leanpub and Coursera. It also performs checking of quiz formatting to ease the quiz upload process to Leanpub and converts Leanpub quizzes to a format ready to upload to Coursera. Without OTTR and *ottrpal*, these formatting changes must be done by authors, which requires knowledge of the specifics of each platform.

3.2 Templates

3.2.1 Quick Start Template Files—The OTTR repository itself is a GitHub template repository, which means that it can be used to create a new repository but this new repository will have its own fresh history. It contains R Markdown templates that demonstrate examples of writing content as well as pull request templates, and issue templates to help facilitate discussions. The pull request template also has a check list to help guide the incorporation of new content.

Upon creation of a new repository based on the OTTR template repository, GitHub issues are automatically filed to help course creators follow along to set up their repository. GitHub issues track the to-do list OTTR-users can follow for the initial set up.

3.2.2 Authorship and Credits Template—Course creation often involves a lot of individuals and the tasks accomplished do not generally fit into the typical authoring system used by academia. Moreover, the existing system for crediting authors in scholarly work has been widely criticized for opacity and lack of accountability and/or deserved credit (Borenstein & Shamoo 2015). We have devised a templated system of credit labels for course contributors which are reported in a table in the course. This credit system is analogous to a film credit system where one individual may have had multiple roles, and roles are shown in an order of highest to lowest level of involvement. We also gained inspiration from the “CRediT” taxonomy.

3.2.3 Course Feedback Templates—In order to make sure a course is properly serving its learners, it needs a mechanism for learners to report concerns and general feedback. On each platform, we have incorporated a templated system that links to a course creator’s chosen method of feedback, such as via GitHub issues or anonymous Google Forms. This way, no matter how a learner chooses to take the course, there is always a readily available mechanism for providing feedback, and in turn, provide continuous pathways for course improvement.

3.3 Detailed Guides

Where automation or additional tools are not reasonable avenues, we have opted to document the process and choices. This guide covers the set up of the repositories and the platforms. It also provides advice and tips for course creation and available tools. Upon usage of the GitHub template and the creation of a new course repository, the automatic issues filed link to this guide in order to help the course creators track and follow along with the guide for set up.

OTTR course development will inevitably involve revisions and debugging. To this end, we have documented the “most common errors” and also encourage individuals to follow our Google Group so that the OTTR course writing community can help.

3.4 OTTR Version Updates

As OTTR is updated and new features are added, changes to the mechanics can be incorporated into downstream courses that utilize the OTTR template (updates are an opt-in system at this point in time). When OTTR has a new release, this change is offered to downstream repositories that have opted-in for updates. These changes are offered in the form of a pull request that course authors can feel free to incorporate as they see fit. These new OTTR feature pull requests uses the repository file sync GitHub action (Schiller 2021). This allows us to reasonably let course creators take advantage of new OTTR features on their course repositories that predated the initial creation of the course repository.

4 Examples

Thus far, 15 courses across 3 institutions, each with multiple contributors (up to 5) have been created with OTTR repository template. Many of these have resulted in courses published on Leanpub and Coursera. By using the OTTR system, the maintenance workload for these updates has been drastically reduced. Figure 6 shows a comparison of the steps involved in the case of a typical wording update to a course. OTTR simplifies updates, making maintenance and minor fixes to courses much more likely to be completed. In the long run, this should help prolong the relevance of these courses.

One such OTTR course regarding Documentation and Usability for Cancer Informatics is shown here as an example (Savonen 2021). This course's own GitHub repository was created from the OTTR Template repository. Figure 7 illustrates what an OTTR course looks like in development as well as on each platform. Edits to the course were done in a text editor. When these changes were added to a pull request they were reviewed and eventually made live. Updates were then automatically rendered on GitHub to a Bookdown website. This rendering was then used to create a Coursera course and Leanpub course that are available for certification. Course updates have been made since its original publication with OTTR these updates have taken less manual steps. Additionally, this course's GitHub repository has received and incorporated updates and new features from the main OTTR repository.

5 Discussion

OTTR is open-source and reduces maintenance burdens. It allows authors to write content once and publish with Bookdown, Leanpub, and Coursera. Menial tasks of course creation are automated by OTTR. It is modular and customize-able, allowing authors to choose what features are most useful for their course. We fully encourage users to utilize any functionality of OTTR that best suits their needs and to bypass functionality their situation does not call for. One such example of OTTR's adaptability is the central configuration file, which allows users to choose which functionality they would like to use. OTTR utilizes the GitHub workflow familiar to many tool developers which encourages collaborative review and transparency. This workflow also allows for community contributions. New OTTR features and changes are offered to course authors through pull requests.

Thus far, OTTR has reduced our own team's burden for creating courses by allowing us to focus more of our time on content. Overall, we believe the creation of OTTR will contribute to greater content creator efficiency, fewer maintenance headaches, and more productive, collaborative, and accessible learning materials in informatics, data science, and beyond.

5.1 Future directions

Where feasible, we will continue to add tools, guidance, and documentation to make course creation and maintenance even less of a burden.

- **Increase usability** - We plan to conduct more usability testing to increase the usability of OTTR so that users can create courses with even more ease.
- **Expand features** - We aim to expand features so OTTR is applicable to a greater variety of course contexts and the needs that accompany them. One such feature includes adding support for how to add OTTR functionality to already existing GitHub repositories that contain R Markdown files.
- **Add guidance** - We intend to add guidance in the form of documentation and videos to help users of OTTR overcome any prerequisite knowledge barriers.
- **Build a community** - We hope to build a community of course creators who can support each other's education endeavors with OTTR tools.

OTTR is a constantly developing set of tools and resources since it is being used and developed by our team and others as we create more course content for our own projects. We greatly value feedback from users of OTTR about how it can be improved. Ideas and feedback can be provided in the form of GitHub issues. There's also a Google Group that we encourage users of OTTR to join.

For more information about OTTR go to ottrproject.org.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

We thank Ira Gooding for his invaluable advice regarding the Leanpub and Coursera platforms.

Funding

This work was supported by the National Cancer Institute under Grant UE5CA254170 and the National Human Genome Research Institute under Grant U24HG010263.

References

Attwood TK, Blackford S, Brazas MD, Davies A & Schneider MV (2017), 'A global perspective on evolving bioinformatics and data science training needs', Briefings in Bioinformatics 20(2), 398–404. **URL:** 10.1093/bib/bbx100**URL:**

Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, Bretaudeau A, Brillet-Guéguen L, Čech M, Chilton J, Clements D, Doppelt-Azeroual O, Erxleben A, Freeberg MA, Gladman S, Hoogstrate Y, Hotz H-R, Houwaart T, Jagtap P, Larivière D, Corguillé GL, Manke T, Mareuil

F, Ramírez F, Ryan D, Sigloch FC, Soranzo N, Wolff J, Videm P, Wolfien M, Wubuli A, Yusuf D, Taylor J, Backofen R, Nekrutenko A & Grüning B (2018), 'Community-driven data analysis training for biology', Cell Systems 6(6), 752–758.e1. **URL:** 10.1016/j.cels.2018.05.012 [PubMed: 29953864] **URL:**

Boettiger C & Eddelbuettel D (2017), 'An Introduction to Rocker: Docker Containers for R', The R Journal 9(2), 527–536. **URL:** 10.32614/RJ-2017-065**URL:**

Borenstein J & Shamoo AE (2015), 'Rethinking authorship in the era of collaborative research', Accountability in Research 22(5), 267–283. [PubMed: 25928178]

Coursera (2021), 'Coursera | Online Courses & Credentials From Top Educators'. **URL:** https://www.coursera.org/**URL:**

Dabbish L, Stuart C, Tsay J & Herbsleb J (2012), Social coding in GitHub, *in* 'Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12', ACM Press. **URL:** 10.1145/2145204.2145396**URL:**

Dabbish L, Stuart C, Tsay J & Herbsleb J (2013), 'Leveraging transparency', IEEE Software 30(1), 37–43. **URL:** 10.1109/ms.2012.172**URL:**

Hemming J, Eide K, Harwood E, Ali R, Zhu Z & Cutler J (2019), 'Exploring professional development for new investigators underrepresented in the federally funded biomedical research workforce.', Ethnicity & Disease 29(Suppl 1), 123–128. [PubMed: 30906160]

Kaiser J (2016), 'Funding for key data resources in jeopardy', Science 351(6268), 14–14. **URL:** 10.1126/science.351.6268.14 [PubMed: 26721983] **URL:**

Kross S, Peng RD, Caffo BS, Gooding I & Leek JT (2020), 'The democratization of data science education', The American Statistician 74(1), 1–7.

Leanpub (2021). **URL:** https://leanpub.com/**URL:**

Mangul S, Mosqueiro T, Abdill RJ, Duong D, Mitchell K, Sarwal V, Hill B, Brito J, Littman RJ, Statz B, Lam AK-M, Dayama G, Grieneisen L, Martin LS, Flint J, Eskin E & Blekhman R (2019), 'Challenges and recommendations to improve installability and archival stability of omics computational tools', bioRxiv . **URL:** https://www.biorxiv.org/content/early/2019/05/14/452532**URL:**

Merkel D (2014), 'Docker: lightweight linux containers for consistent development and deployment', Linux journal 2014(239), 2.

Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T & Evans BD (2020), 'Ten simple rules for writing dockerfiles for reproducible data science', PLOS Computational Biology 16(11), 1–24.

Rivas MJ, Baker RB & Evans BJ (2020), 'Do MOOCs make you more marketable? an experimental analysis of the value of MOOCs relative to traditional credentials and experience', AERA Open 6(4), 233285842097357. **URL:** 10.1177/2332858420973577**URL:**

RStudio (2022), rmarkdown: Dynamic Documents for R. R package version 2.11.15. **URL:** https://github.com/rstudio/rmarkdown**URL:**

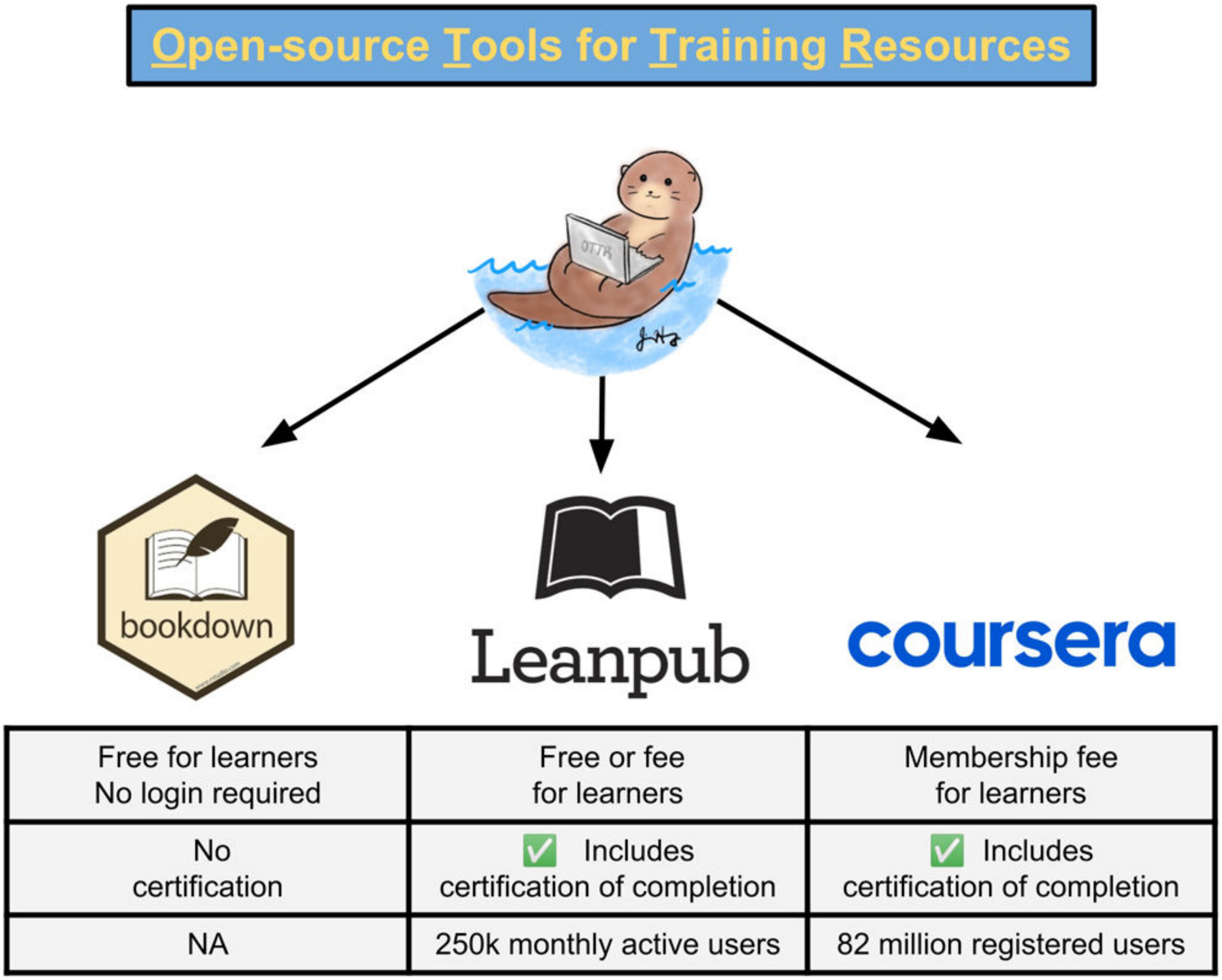
Savonen C (2021), 'Documentation and usability'. **URL:** https://bit.ly/docs-n-usability**URL:**

Schiller M (2021), 'Repo File Sync Action - GitHub Marketplace'. **URL:** https://github.com/marketplace/actions/repo-file-sync-action**URL:**

Wilson G (2016), 'Software carpentry: lessons learned', F1000Research 3, 62. **URL:** 10.12688/f1000research.3-62.v2**URL:**

Wren JD (2016), 'Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades', Bioinformatics 32(17), 2686–2691. **URL:** 10.1093/bioinformatics/btw284 [PubMed: 27153671] **URL:**

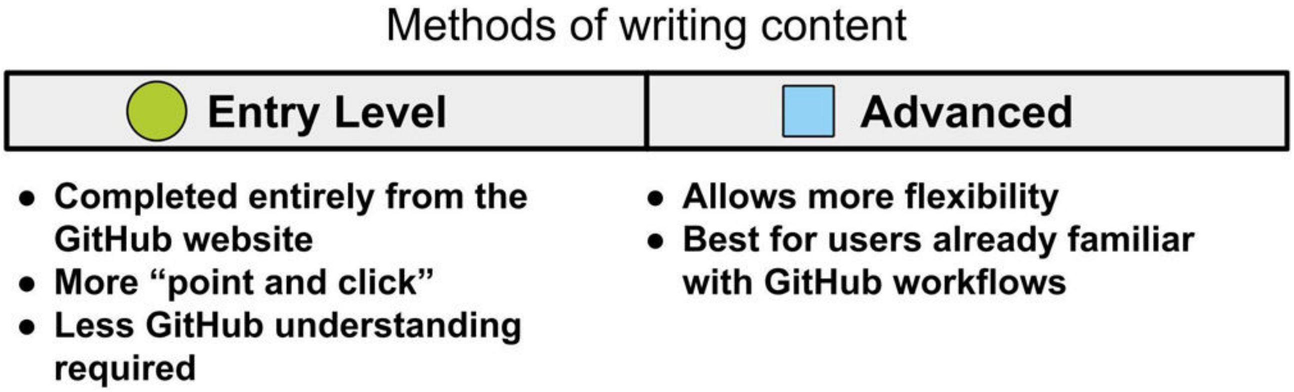
Xie Y (2016), bookdown: Authoring Books and Technical Documents with R Markdown, Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1138700109. **URL:** https://bookdown.org/yihui/bookdown**URL:**



Free for learners No login required	Free or fee for learners	Membership fee for learners
No certification	✅ Includes certification of completion	✅ Includes certification of completion
NA	250k monthly active users	82 million registered users

Figure 1:

Publishing platform options. OTTR can publish to three platforms which each have distinct audiences and meet different learner needs. The OTTR system minimizes the maintenance pains for upkeep on these three platforms by allowing you to write once and have updates to the material automatically sent to all platforms.



Methods of writing content

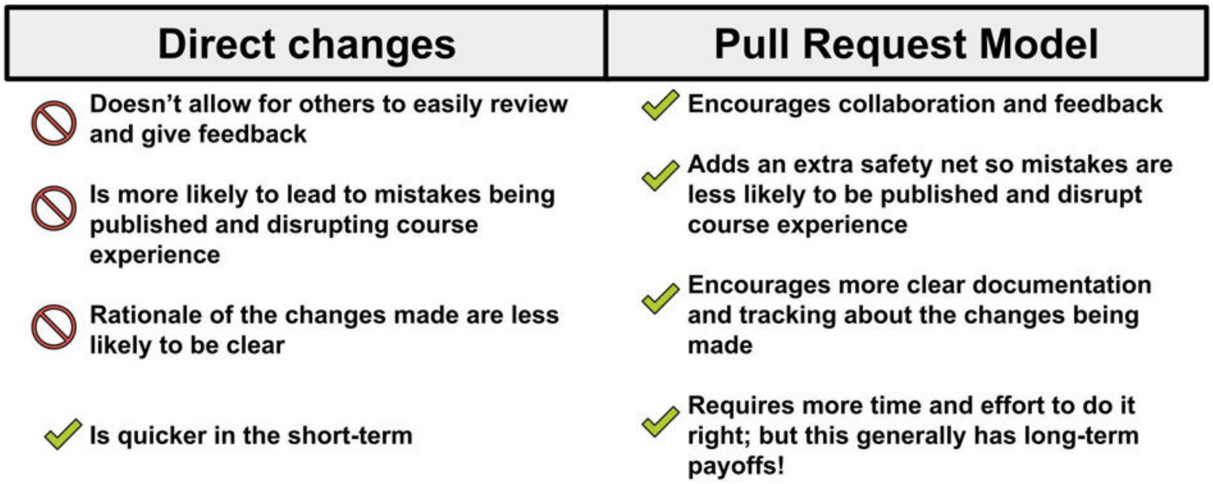
🟢 Entry Level	🟦 Advanced

- Completed entirely from the GitHub website
- More "point and click"
- Less GitHub understanding required

- Allows more flexibility
- Best for users already familiar with GitHub workflows

Figure 2:

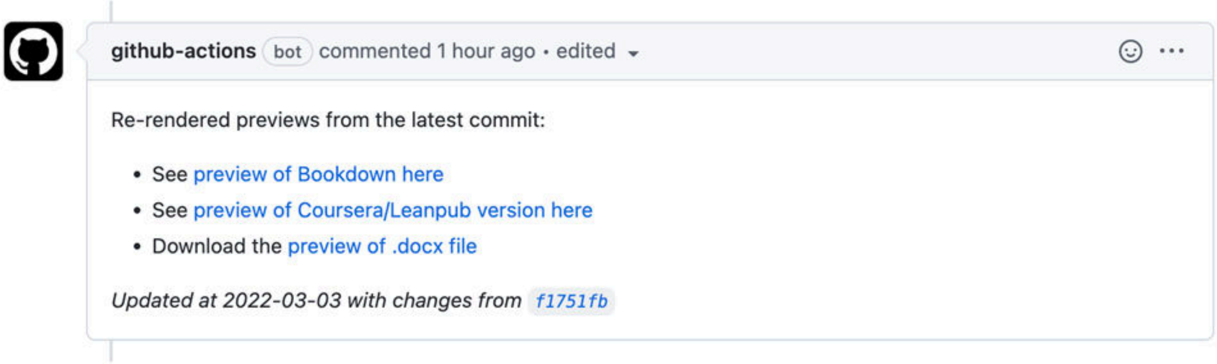
OTTR supports two writing process methods discussed in our documentation. Users can choose based on their familiarity and interest in using GitHub.



Direct changes	Pull Request Model
🚫 Doesn't allow for others to easily review and give feedback	✓ Encourages collaboration and feedback
🚫 Is more likely to lead to mistakes being published and disrupting course experience	✓ Adds an extra safety net so mistakes are less likely to be published and disrupt course experience
🚫 Rationale of the changes made are less likely to be clear	✓ Encourages more clear documentation and tracking about the changes being made
✓ Is quicker in the short-term	✓ Requires more time and effort to do it right; but this generally has long-term payoffs!

Figure 3:

OTTR is based on using the GitHub pull request development model. The pull request allows collaborative development on content that leads to more polished changes before publishing. It does require more time upfront than directly changing content. However, in the long-term this pull request model generally prevents errors, encourages collaboration, and increases content transparency.

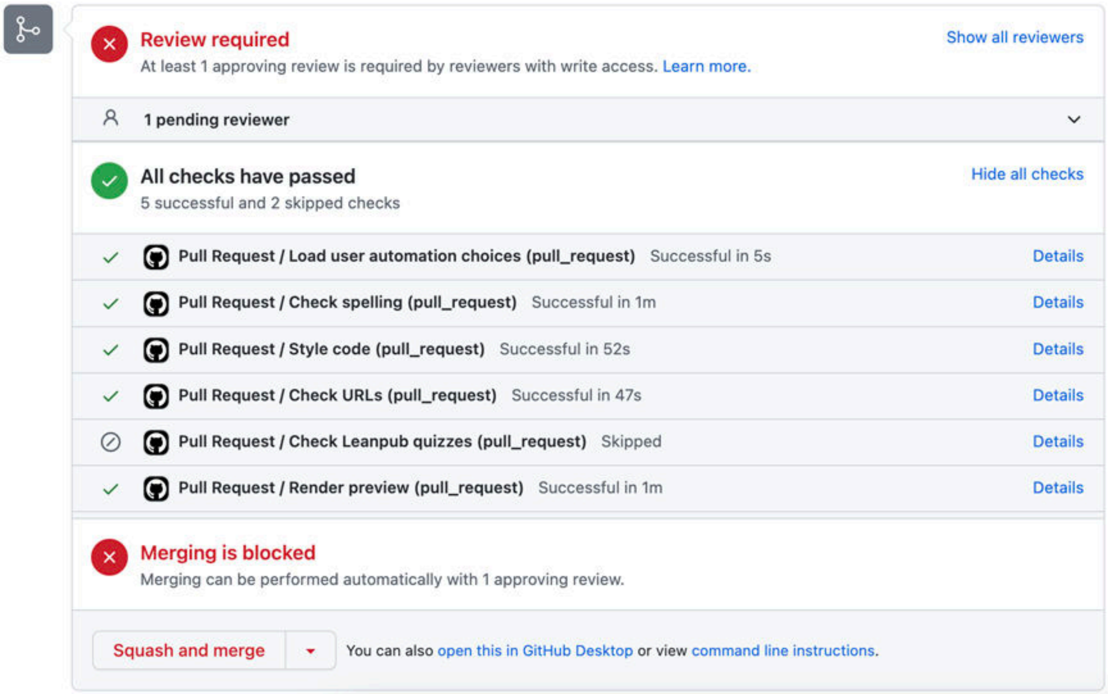


Re-rendered previews from the latest commit:

- See preview of Bookdown here
- See preview of Coursera/Leanpub version here
- Download the preview of .docx file

Updated at 2022-03-03 with changes from `f1751fb`

Figure 4:
Automatic rendering. Upon opening a pull request, OTTR creates a preview of the changes in different formats so the content creator can easily review the changes. The preview is provided in three ways: A link to what the Bookdown rendered website would look like, a link to what the Coursera/Leanpub material iframe preview would look like, and a docx file which allows for use of track changes with Google Docs or Microsoft Word.



✕ **Review required** — Show all reviewers

At least 1 approving review is required by reviewers with write access. Learn more.

1 pending reviewer

✓ **All checks have passed** — Hide all checks

5 successful and 2 skipped checks

✓	**Pull Request / Load user automation choices (pull_request)**	Successful in 5s	Details
✓	**Pull Request / Check spelling (pull_request)**	Successful in 1m	Details
✓	**Pull Request / Style code (pull_request)**	Successful in 52s	Details
✓	**Pull Request / Check URLs (pull_request)**	Successful in 47s	Details
⊘	**Pull Request / Check Leanpub quizzes (pull_request)**	Skipped	Details
✓	**Pull Request / Render preview (pull_request)**	Successful in 1m	Details

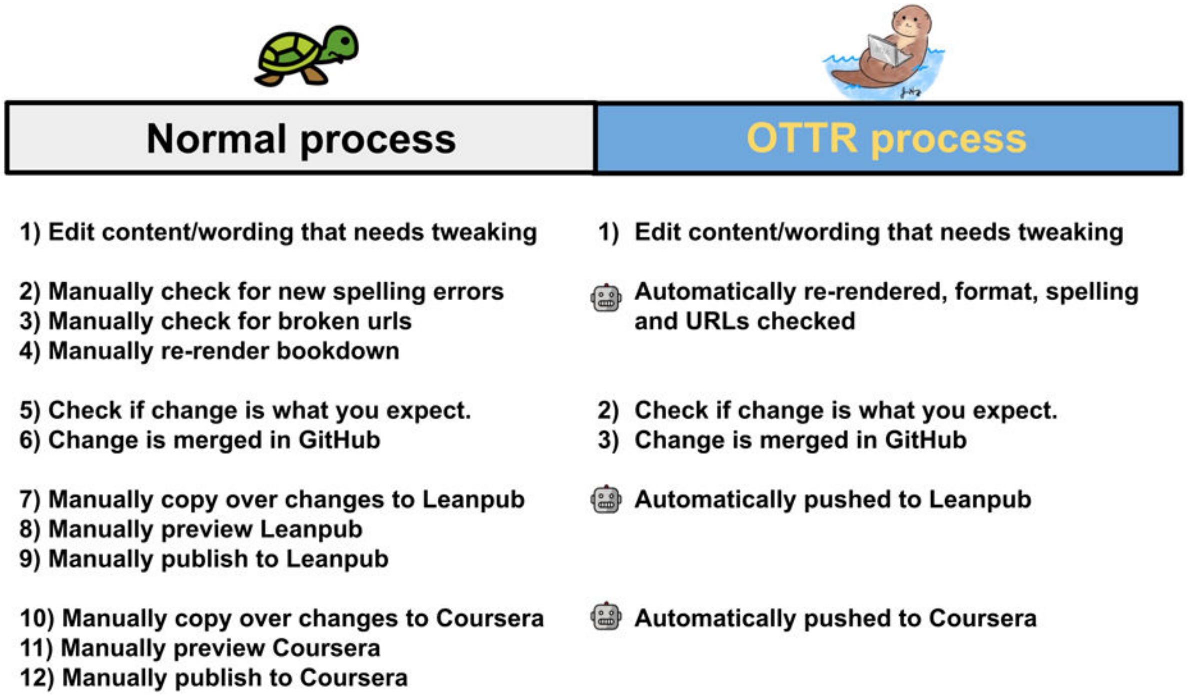
✕ **Merging is blocked**

Merging can be performed automatically with 1 approving review.

You can also open this in GitHub Desktop or view command line instructions.

Figure 5:

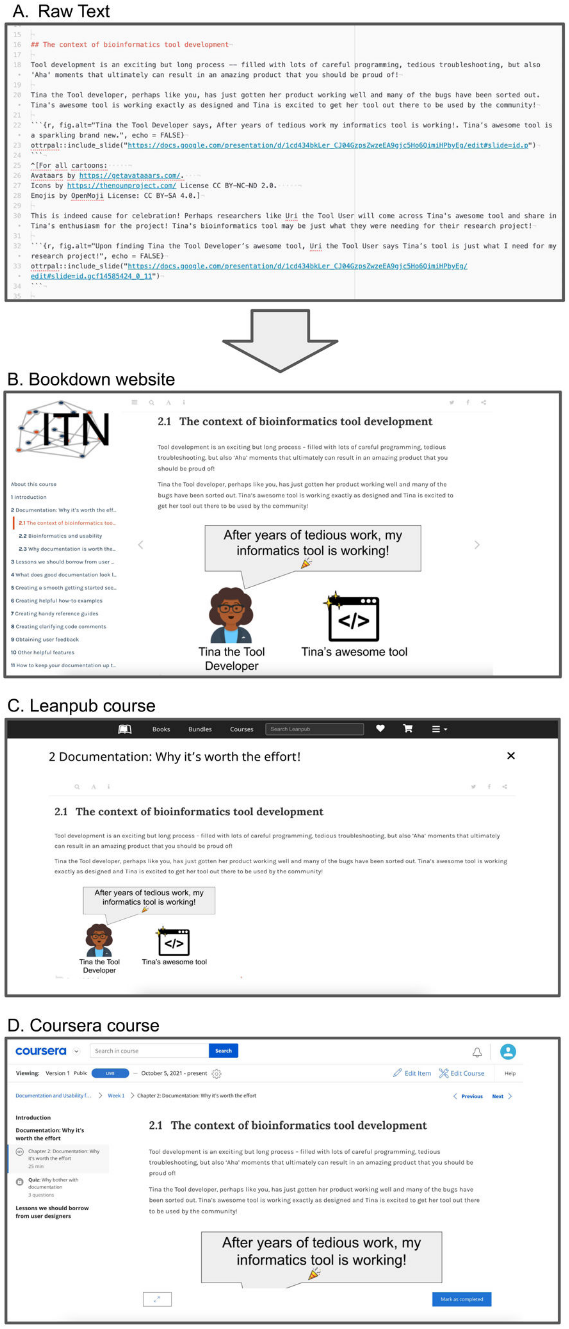
Automatic checks. Upon opening a pull request, OTTR runs a series of customize-able checks on the new content. These can be turned on or off by a central configuration file.



Normal process	OTTR process
1) Edit content/wording that needs tweaking	1) Edit content/wording that needs tweaking
2) Manually check for new spelling errors 3) Manually check for broken urls 4) Manually re-render bookdown	🤖 Automatically re-rendered, format, spelling and URLs checked
5) Check if change is what you expect. 6) Change is merged in GitHub	2) Check if change is what you expect. 3) Change is merged in GitHub
7) Manually copy over changes to Leanpub 8) Manually preview Leanpub 9) Manually publish to Leanpub	🤖 Automatically pushed to Leanpub
10) Manually copy over changes to Coursera 11) Manually preview Coursera 12) Manually publish to Coursera	🤖 Automatically pushed to Coursera

Figure 6:

Maintenance steps with and without OTTR. Note that the steps with a robot icon are done automatically in the OTTR setup. The checks performed by OTTR reduce the time and mental energy it takes make course updates. Ideally, courses should be regularly updated, meaning the overall burden of course maintenance is multiplied every time updates to the courses are done. Over the life of the course, OTTR saves course creators time and mental energy.



~~~
## The context of bioinformatics tool development

Tool development is an exciting but long process -- filled with lots of careful programming, tedious troubleshooting, but also 'Aha' moments that ultimately can result in an amazing product that you should be proud of!

Tina the Tool developer, perhaps like you, has just gotten her product working well and many of the bugs have been sorted out. Tina's awesome tool is working exactly as designed and Tina is excited to get her tool out there to be used by the community!

```{r, fig.alt="Tina the Tool Developer says, After years of tedious work my informatics tool is working!. Tina’s awesome tool is a sparkling brand new.", echo = FALSE}
ottrpal::include_slide("https://docs.google.com/presentation/d/1cd434bkLer_CJ04GzpsZwzeEA9gjc5Ho6QimiHPbyEg/edit#slide=id.p")
~~~
^[For all cartoons:
Avataars by https://getavataaars.com/.
Icons by https://thenounproject.com/ License CC BY-NC-ND 2.0.
Emojis by OpenMoji License: CC BY-SA 4.0.]

This is indeed cause for celebration! Perhaps researchers like Uri the Tool User will come across Tina's awesome tool and share in Tina's enthusiasm for the project! Tina's bioinformatics tool may be just what they were needing for their research project!

```{r, fig.alt="Upon finding Tina the Tool Developer’s awesome tool, Uri the Tool User says Tina’s tool is just what I need for my research project!", echo = FALSE}
ottrpal::include_slide("https://docs.google.com/presentation/d/1cd434bkLer_CJ04GzpsZwzeEA9gjc5Ho6QimiHPbyEg/edit#slide=id.gcf14585424_0_11")
```
```

B. Bookdown website
ITN
About this course
1 Introduction
2 Documentation: Why it's worth the eff...
2.1 The context of bioinformatics too...
2.2 Bioinformatics and usability
2.3 Why documentation is worth the...
3 Lessons we should borrow from user ...
4 What does good documentation look l...
5 Creating a smooth getting started sec...
6 Creating helpful how-to examples
7 Creating handy reference guides
8 Creating clarifying code comments
9 Obtaining user feedback
10 Other helpful features
11 How to keep your documentation up t...

2.1 The context of bioinformatics tool development

Tool development is an exciting but long process – filled with lots of careful programming, tedious troubleshooting, but also ‘Aha’ moments that ultimately can result in an amazing product that you should be proud of!

Tina the Tool developer, perhaps like you, has just gotten her product working well and many of the bugs have been sorted out. Tina’s awesome tool is working exactly as designed and Tina is excited to get her tool out there to be used by the community!

2 Documentation: Why it’s worth the effort!

2.1 The context of bioinformatics tool development

Tool development is an exciting but long process – filled with lots of careful programming, tedious troubleshooting, but also ‘Aha’ moments that ultimately can result in an amazing product that you should be proud of!

Tina the Tool developer, perhaps like you, has just gotten her product working well and many of the bugs have been sorted out. Tina’s awesome tool is working exactly as designed and Tina is excited to get her tool out there to be used by the community!

After years of tedious work, my informatics tool is working!
Tina the Tool Developer
Tina’s awesome tool
D. Coursera course
coursera
Search in course
Search
Viewing: Version 1 Public LIVE — October 5, 2021 - present
Edit Item
Edit Course
Help
Documentation and Usability f... > Week 1 > Chapter 2: Documentation: Why it's worth the effort
Previous
Next
Introduction
Documentation: Why it's worth the effort
Chapter 2: Documentation: Why it's worth the effort
25 min
Quiz: Why bother with documentation
3 questions
Lessons we should borrow from user designers

2.1 The context of bioinformatics tool development

Tool development is an exciting but long process – filled with lots of careful programming, tedious troubleshooting, but also ‘Aha’ moments that ultimately can result in an amazing product that you should be proud of!

Tina the Tool developer, perhaps like you, has just gotten her product working well and many of the bugs have been sorted out. Tina’s awesome tool is working exactly as designed and Tina is excited to get her tool out there to be used by the community!

After years of tedious work, my informatics tool is working!
Mark as completed

Figure 7:

Example of rendered content. The first panel (A) shows the raw text for this example course page made with OTTR. It shows how the raw text written by a content author might look like as (B) a Bookdown website hosted by GitHub pages, (C) a Leanpub course and (D) Coursera course. This automatic rendering has an added benefit of keeping the new file changes focused on the content changes instead of html rendering artifacts.
```