



# Leaving reviews and responding to reviews

Childhood Cancer  
**Data**  **Lab**

x





# Tips for leaving PR reviews



# When leaving a review...

Start by looking through the linked issue and any associated discussion

As needed, familiarize yourself with relevant code *before* the PR code

- If you are entirely unfamiliar with the code, maybe you're not the best person to review this!

Read the PR comment carefully, including any suggestions from the author for where/how you should focus your review

- This may help you determine how to perform review: Just read through on GitHub, locally checkout out the branch and test the code, or a little of both!
- Depending on the circumstance, you may want to leave specific in-line comments or overall high-level comments, or a little of both!





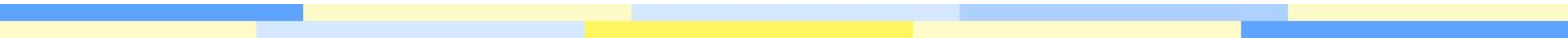
# Tips for responding to PR reviews



# Start by taking in the *big picture*

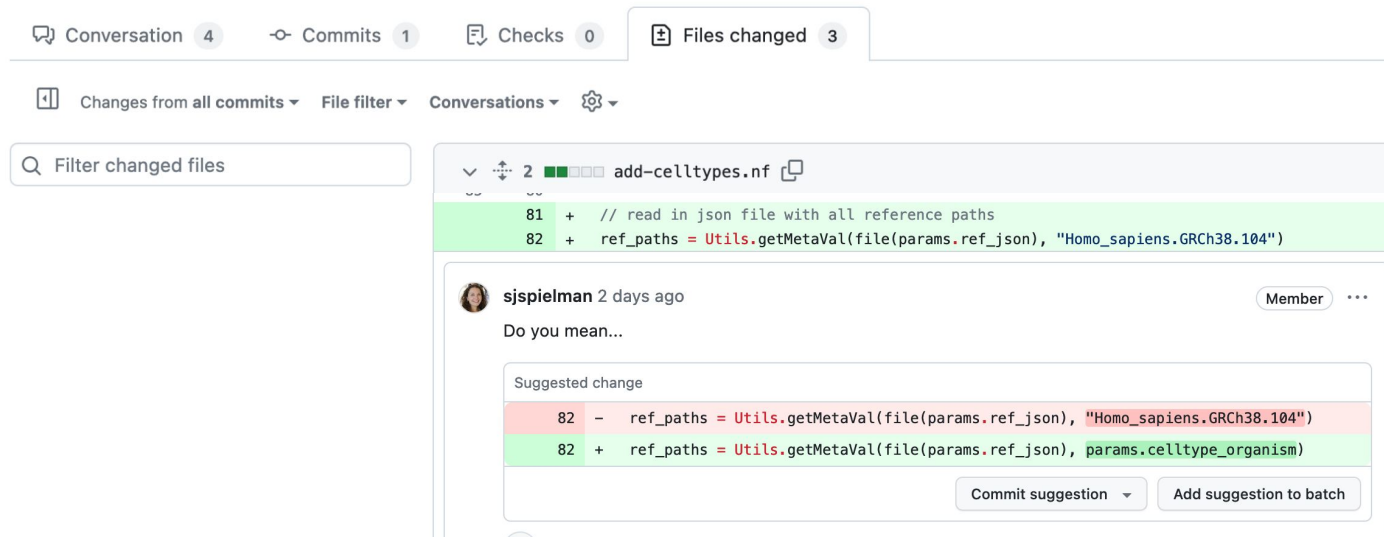
It's tempting to dive right in and respond to individual comments, but you'll gain more context about why the reviewer left certain comments by reading the whole review first

- For example, maybe that the reviewer misunderstood something you were trying to do, which *might happen if* you didn't clearly express the goals you had in mind for the review
- By taking in the full review, you might identify other areas to improve that weren't specifically reviewed, but conceptually mentioned



# If your reviewer left in-line suggestions...

You can either accept suggestions directly or add then to a batch of changes to commit all at once.



The screenshot shows a code review interface. At the top, there are tabs for 'Conversation' (4), 'Commits' (1), 'Checks' (0), and 'Files changed' (3). Below these are filters for 'Changes from all commits', 'File filter', 'Conversations', and a settings icon. A search bar on the left contains the text 'Filter changed files'. The main area displays a file named 'add-celltypes.nf' with two lines of code: line 81 is a comment '// read in json file with all reference paths', and line 82 is 'ref\_paths = Utils.getMetaVal(file(params.ref\_json), "Homo\_sapiens.GRCh38.104")'. A review comment from 'sjspielman' 2 days ago asks 'Do you mean...'. Below the comment, a 'Suggested change' box shows a diff: line 82 is changed from 'ref\_paths = Utils.getMetaVal(file(params.ref\_json), "Homo\_sapiens.GRCh38.104")' to 'ref\_paths = Utils.getMetaVal(file(params.ref\_json), params.celltype\_organism)'. At the bottom of the suggestion box are two buttons: 'Commit suggestion' and 'Add suggestion to batch'.


Conversation 4 Commits 1 Checks 0 Files changed 3

Changes from all commits File filter Conversations

Filter changed files

2 add-celltypes.nf

```
81 + // read in json file with all reference paths
82 + ref_paths = Utils.getMetaVal(file(params.ref_json), "Homo_sapiens.GRCh38.104")
```

 sjspielman 2 days ago Member ...

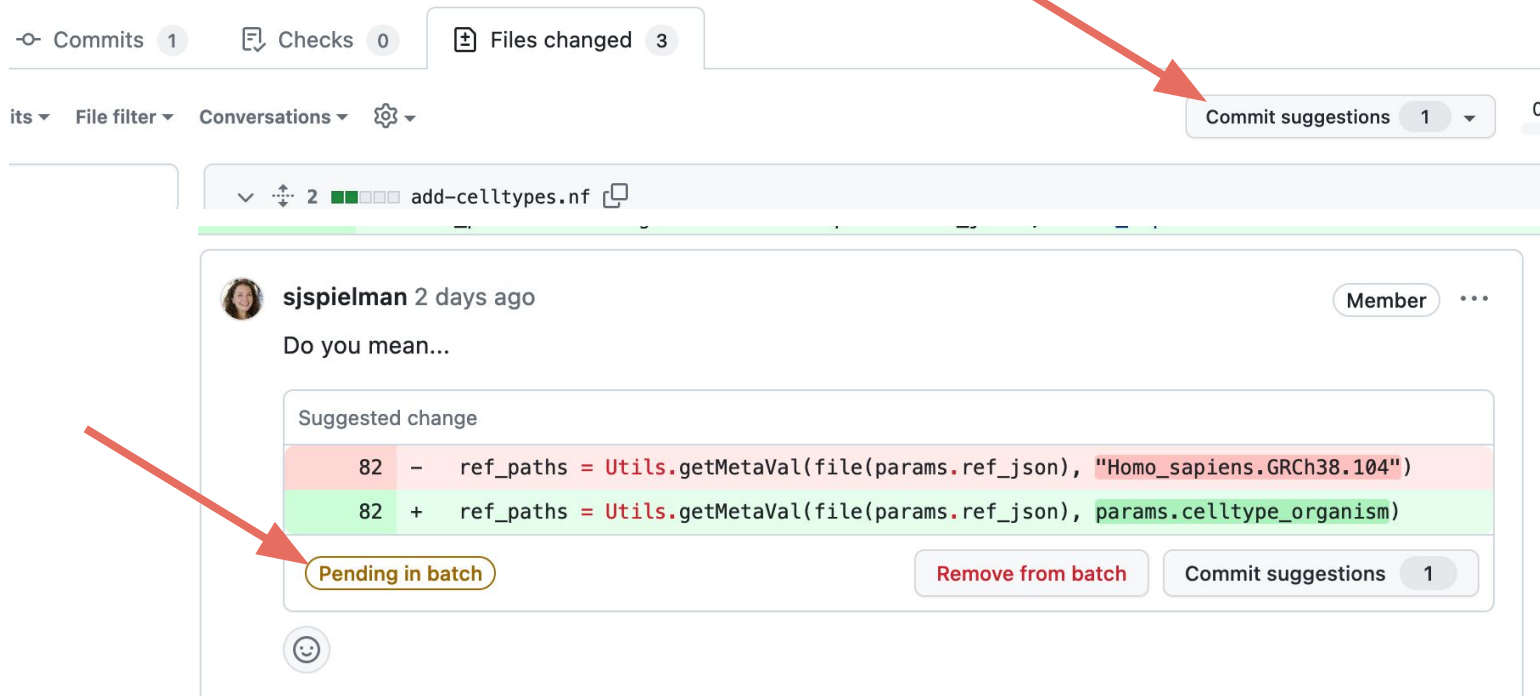
Do you mean...

Suggested change

```
82 - ref_paths = Utils.getMetaVal(file(params.ref_json), "Homo_sapiens.GRCh38.104")
82 + ref_paths = Utils.getMetaVal(file(params.ref_json), params.celltype_organism)
```

Commit suggestion Add suggestion to batch

# When all suggestions are added, commit them all at once



The screenshot shows a GitHub pull request interface. At the top, there are tabs for 'Commits 1', 'Checks 0', and 'Files changed 3'. Below these, there are filters for 'its', 'File filter', 'Conversations', and a settings icon. On the right side, there is a 'Commit suggestions 1' button with a dropdown arrow. A red arrow points from the top right towards this button. Below the filters, there is a file path 'add-celltypes.nf' with a copy icon. The main content area shows a comment from 'sjspielman 2 days ago' with the text 'Do you mean...'. Below the comment, there is a 'Suggested change' box. Inside this box, there is a diff showing a change on line 82. The original code is 'ref\_paths = Utils.getMetaVal(file(params.ref\_json), "Homo\_sapiens.GRCh38.104")' and the suggested change is 'ref\_paths = Utils.getMetaVal(file(params.ref\_json), params.celltype\_organism)'. A red arrow points from the left towards the 'Pending in batch' button at the bottom of the suggested change box. To the right of the 'Pending in batch' button are two other buttons: 'Remove from batch' and 'Commit suggestions 1'.

Commits 1 Checks 0 Files changed 3

its File filter Conversations

Commit suggestions 1

add-celltypes.nf

sjspielman 2 days ago Member

Do you mean...


Suggested change

```
82 - ref_paths = Utils.getMetaVal(file(params.ref_json), "Homo_sapiens.GRCh38.104")
82 + ref_paths = Utils.getMetaVal(file(params.ref_json), params.celltype_organism)
```

Pending in batch Remove from batch Commit suggestions 1

# Suggestions are fantastic, but *always check them*

Suggestions are usually not pieces of tested code - you will want to make sure they actually work as expected when responding to review (aka, run the suggested code!)

 **Caution!** Accepting a suggestion will resolve the review comment, which may not be what you want to do!

- As a reviewer: limit suggestion comments to suggestions, and leave other comments elsewhere



# Making changes to a pull request

- Check out the branch for the PR that is under review
- Pull any changes you made in GitHub
- Make any changes in response to review and *test* them as needed
  - Keep the individual commits appropriately sized!
- Push your changes back to GitHub
  - All changes will automatically be reflected as updates to the PR



# When re-requesting review...

Don't *just* re-request review - communicate with your reviewer!


- Summarize the changes you made in an overall comment, including any changes you made that the reviewer did not request
  - Optional: Communicate specific changes you made by sharing the commit hash
- Maybe there were some requested changes you *didn't* make - explain why, or ask the reviewer for more clarification
- Your reviewer is on your side, don't take comments personally! It's all room for growth

# Make your reviewer's life a little easier

Tell them where/when you addressed their comment


- GitHub will automatically style and link commit hashes


```
bin/post_process_sce.R Outdated
103 - }
104 - metadata(sce)$scpca_filter_method <- paste0(metadata(sce)$scpca_filter_method,
105 -                                           adt_filter_string)
92 + metadata(sce)$scpca_filter_method <- metadata(sce)$scpca_filter_method
```

 **jashapiro** on Jun 2 Member ...

What is meant to be happening here?


I *think* what we want here is an accounting of the adt filter method ( `adt_scpca_filter_method` ?) which should reflect whether negative controls were used.




 **sjspielman** on Jun 2 Member Author ...

What is happening here is I had to remove `paste` code but then apparently did not look at the resulting line to see...we don't need this line at all!

But yes, let's add `adt_scpca_filter_method` as a separate metadata "cleanTagCounts".



 **sjspielman** on Jun 2 Member Author ...

- Silly line removed in [39d3a43](#)
- Added adt metadata in [18783e9](#)

# You can open new issues based on PR comments

- Keeps the project moving along - otherwise, we could be here *a while*
- Keeps each PR in a reasonable scope

templates/qc\_report/cite\_qc.rmd

```
42 - arrange(desc(mean)) |>
42 + # ensure `target` is the first level of target_type
43 + mutate(target_type = forcats::fct_relevel(target_type, "target")) |>
44 + arrange(target_type, desc(mean)) |>
```



jashapiro last month

Member ...

Oh, one more comment, which I don't want to make hold this up: When we have a lot of ADTs, this table gets kind of unreasonably long. We could limit this to a set number of ADTs, or collapse the table, but I'm not really sure what the best option is here. Maybe a JS table like <https://glin.github.io/reactable/> ? This would be a whole separate thing, and I don't think it is worth doing in this PR, or for this release.



sjspielman last month

Member Author ...

When we have a lot of ADTs, this table gets kind of unreasonably long.

yes, *yes it does*.

I'll open an issue for circling back to this.



## Update QC report tables #371

Open sjspielman opened this issue last month · 0 comments



sjspielman commented last month

Member ...

From this [comment](#)

Oh, one more comment, which I don't want to make hold this up: When we have a lot of ADTs, this table gets kind of unreasonably long. We could limit this to a set number of ADTs, or collapse the table, but I'm not really sure what the best option is here. Maybe a JS table like <https://glin.github.io/reactable/> ?

We should take some time to revisit tables overall, including but not limited to excessively long tables. Another lightweight (aka no additional dependencies) approach could be `DT::datatable()`, but we should probably explore a couple options.



# Don't rush to merge when PR has been approved

You still need to read the review comments! For example, depending on a team's policy/culture, it may be acceptable for a reviewer to approve **under the assumption you'll make a couple more small changes**.



allyhawkins approved these changes 2 weeks ago

[View reviewed changes](#)

allyhawkins left a comment

Member ...

This looks much cleaner! I just have a few minor comments, but I don't need to see it again.



R/calculate\_within\_batch\_ari.R

Outdated

Show resolved

R/calculate\_within\_batch\_ari.R

Outdated

Show resolved

R/calculate\_within\_batch\_ari.R

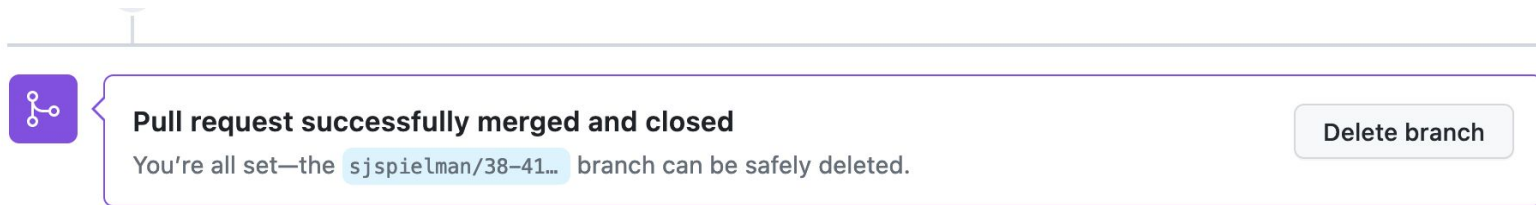
Outdated

Show resolved

The "few minor comments"

# Don't forget to delete your branch after merging!

*Especially* if the branch was stacked!



(Note that this won't delete your *local* copy of the branch.)