What do I mean by **system approaches**?

Anything that goes beyond the scope of an individual pull request (i.e., applies to all of them) or an individual author's responsibilities.

Alt title: Solving pull request problems at a deeper level!

Let's imagine a pull request where we want to make sure:

- All the packages required to run the new code are in the project environment

- The new code can be reused on other data sets

- The approach is methodologically sound

❌ Dependencies won't install ⇒

✅ Dependencies installed

❌ Code is not robust to underlying data changes ⇒

✅ Dependencies installed

✅ Code runs on test data

❌ Reviewer points out incorrect simplifying assumption

Better done by robots 🤖

Needs people

If we can learn to **spot tasks for robots**, we can **save reviewers' energy** for the things that need it

We can also leverage automations to codify and enforce policies.
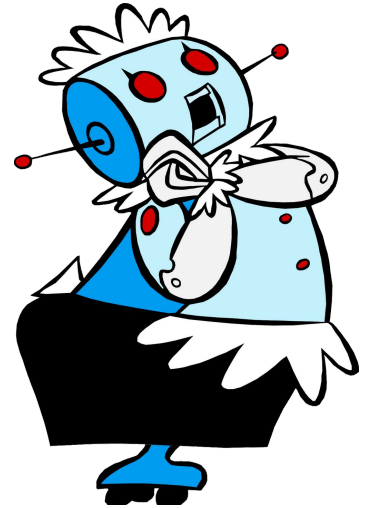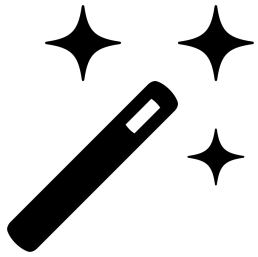
Beyond automation, we can use other features of GitHub to **create structure** that appears when people perform the task of **creating a pull request**.

Welcoming our robot overlords

A **GitHub Actions** primer

GitHub Actions (GHA) is a service through GitHub that allows for automation of workflows. GitHub and third parties create "actions," which are like template workflows (e.g., checking out a repository or pushing to Dockerhub).

The nice part about GHA is that it can be triggered on many different GitHub events like a commit to a pull request targeting a specific branch or when an issue is created.

Energy-saving example: Spell check

Using GitHub Actions to check for spelling errors – failing if any errors are detected – allows pull request authors to correct them before a reviewer looks at it.

Let's take a look at some real world examples:

`training-modules` workflow, Rscript, and custom dictionary

`refinebio-docs` workflow, config, and custom dictionary

Codifying policies example: Styling

Using GitHub Actions to enforce code style relieves authors' and reviewers' burden of styling code and enforcing style, respectively.

Debate about style can happen once – when selecting or implementing the workflow – instead of on every pull request.

Let's take a look at some real world examples:

`refinebio-examples` [workflow](#)

`scpcaTools` [workflow](#)

Helping PR authors fulfill their responsibilities with templates

Pull request templates can help prompt authors to satisfy requirements

Whether that's providing context for the pull request, linking to an issue, or a checklist of required steps

Real world examples of **pull request templates**

In the next few slides, we'll review

- A single, universal PR template from the `OpenPBTA-analysis` project

- Multiple PR templates (and how to use them) from a repository called `refinebio-examples`

Purpose/implementation Section

🔗 **What scientific question is your analysis addressing?**

What was your approach?

What GitHub issue does your pull request address?

Directions for reviewers. Tell potential reviewers what kind of feedback you are soliciting.

Which areas should receive a particularly close look?

Is there anything that you want to discuss further?

Is the analysis in a mature enough form that the resulting figure(s) and/or table(s) are ready for review?

Results

What types of results are included (e.g., table, figure)?

What is your summary of the results?

https://github.com/AlexsLemonade/OpenPBTA-analysis/blob/master/.github/PULL_REQUEST_TEMPLATE.md

Reproducibility Checklist

- [ ] The dependencies required to run the code in this pull request have been added to the project Dockerfile.
- [ ] This analysis has been added to continuous integration.

Documentation Checklist

- [ ] This analysis module has a `README` and it is up to date.
- [ ] This analysis is recorded in the table in `analyses/README.md` and the entry is up to date.
- [ ] The analytical code is documented and contains comments.

https://github.com/AlexsLemonade/OpenPBTA-analysis
/blob/master/.github/PULL_REQUEST_TEMPLATE.md

An HTML comment includes another checklist for data releases

```
<!-- IF YOUR PULL REQUEST IS A DATA RELEASE, PLEASE REMOVE THE [HTML COMMENT TAG](https://html.com/tags/comment-tag/)
FROM THE SECTION BELOW AND COMPLETE THE CHECKLIST-->

<!--
### Data Release Checklist
- [ ] Is the table in doc/data-file-descriptions.md up to date?
- [ ] Is doc/data-format.md up to date?
- [ ] Is doc/release-notes.md up to date?
- [ ] Is download-data.sh up to date?
- [ ] Was download-data.sh tested and did it complete without error?
-->
```

https://github.com/AlexsLemonade/OpenPBTA-analysis
/blob/master/.github/PULL_REQUEST_TEMPLATE.md

One weird trick for **multiple pull request templates**

AlexsLemonade / refinebio-examples

<> Code   Issues 33   Pull requests 1   Zenhub   Discussions   Actions   Projects   Wiki   Security   Insights   ···

staging

refinebio-examples / .github / PULL_REQUEST_TEMPLATE.md

Go to file   t   ···

cansavvy   Make the "Other" PR template the default (#341)   ···   70e58bb · 3 years ago   History

Preview   Code   Blame   33 lines (18 loc) · 1.23 KB   Raw

Use the 'Preview' view to click on a link below to choose an appropriate PR template:

For any stage of adding a new analysis example: New analysis PR

For publishing changes to Github pages: Publishing PR

For either stage of a hotfix PR -- something that is straightforward and needs to be user-facing quickly: Hotfix PR

For any other types of PRs that don't fit any of the above categories, delete the previous section (including this line) and use the template below.

Purpose

Issue addressed

Summary

- GitHub Actions – or other continuous integration/continuous deployment (CI/CD) services – can automate some things we'd like to check during a review, thereby taking them off a human reviewer's plate.

- Pull request templates support individual authors in meeting their responsibilities by reminding them of what to include in an initial pull request comment right when they write it.