Updating the Liu Lab Website via GitHub Pull Requests

Hyun-Hwan Jeong

Liu Lab Meeting

February 3, 2026

Problem: Centralized Website Maintenance

The old way (WordPress):

- Server-based, needs constant maintenance
- One person maintains the website
- Everyone sends updates via email/Slack
- Bottleneck — slow updates
- Single point of failure

The new way (Static Site):

- Compile once, add items gradually
- **You** update your own info via PR
- No server maintenance needed
- No bottleneck
- Shared responsibility

Goal

Each lab member owns and maintains their own profile on the lab website.

The GitHub Workflow Is Everywhere

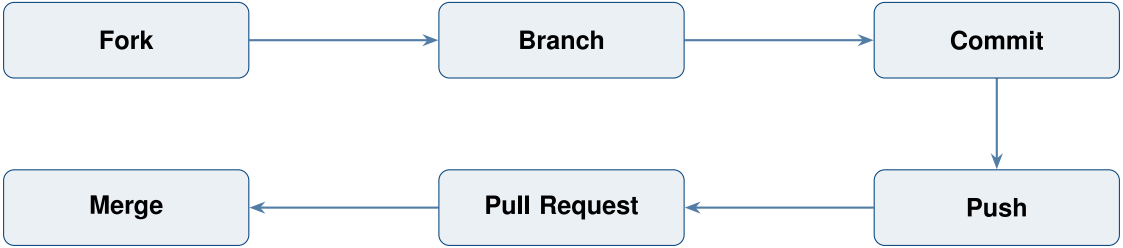
The fork → branch → PR workflow you just learned is **the standard** in:

- **Open-source software** — contributing to any project on GitHub
- **Industry** — how teams collaborate at Google, Meta, startups
- **Research** — collaborative papers, shared analysis pipelines
- **Data science** — reproducible workflows and version control

Transferable skills

By updating the lab website, you practice the **exact same workflow** used to contribute to projects like `scikit-learn`, `pandas`, `pytorch`, etc.

Key Concepts You're Learning



Each step maps to a **real-world collaboration pattern**: isolation (branch), review (PR), integration (merge).

What You Need

Tools

- A **GitHub account**
- `git` installed on your machine
- `gh` — the GitHub CLI (optional)
- A text editor (VS Code, nano, vim, etc.)

Knowledge (optional)

- Basic terminal / command line
- Markdown syntax
- YAML frontmatter (we'll cover this)

Install GitHub CLI

```
# macOS
brew install gh

# Windows
winget install --id GitHub.cli

# Then authenticate
gh auth login
```

Step 1: Fork the Repository

Go to `github.com/LiuzLab/Liuzlab.github.io` and click **Fork**

Or use the CLI:

```
# Fork the repository to your GitHub account
gh repo fork LiuzLab/Liuzlab.github.io --clone
```

This will:

1. Create a copy (**fork**) under your GitHub account
2. Clone it to your local machine
3. Set up `origin` (your fork) and `upstream` (the lab repo)

Step 2: Create a Branch

Never work directly on `main`. Always create a branch:

```
# Navigate into the repo
cd Liuzlab.github.io

# Make sure you're up to date
git checkout main
git pull upstream main

# Create a new branch for your changes
git checkout -b update/your-name-info
```

Branch naming convention

Use descriptive names: `update/jane-smith-info`, `fix/typo-in-bio`, `add/new-member-alex`

The Website Structure (Astro v6)

The lab website is built with **Astro** — a modern static site generator.

Key directories:

- `src/content/team/` — **Member profiles**
- `src/assets/` — Photos & images
- `src/pages/` — Page layouts

File naming convention:

`{2-3 letter role}-{first}-{last}.md`

`prof-`	Professor
`ap-`	Assistant Professor
`pd-`	Postdoc
`phd-`	PhD Student
`ms-`	Master Student
`ug-`	Undergraduate
`ae-`	Alumni/Emeritus

Repository layout

```
Liuzlab.github.io/
  src/
    content/
      team/
        prof-zhandong-liu.md
        ap-hyun-hwan-jeong.md
        phd-emily-chen.md
        ae-johnathan-jia.md
        ...
    assets/
      zhandong-liu.jpg
      hyun-hwan-jeong.jpg
      ...
    pages/
      ...
```

Step 3: Edit Your Profile — Real Example

YAML frontmatter (metadata):

```
---
name: "Hyun-Hwan Jeong"
role: "Assistant Professor"
title: ["Assistant Professor"]
avatar: "../../assets/hyun-hwan-jeong.jpg"
bio: "Computational biology and
  bioinformatics tool development"
email: "hyun-hwan.jeong@bcm.edu"
linkedin: "https://www.linkedin.com/
  in/hyunhwan-jeong/"
github: "https://github.com/hyunhwan-bcm"
googleScholar: "https://scholar.google.
  com/citations?user=RHsMwPUAAAAJ"
weight: 2
---
```

What to fill in:

`name`	**Your full name**
`role`	Pick from the valid list
`avatar`	Path to your photo
`bio`	**One short sentence** about your research
`email`	Your BCM email
`linkedin`	Your profile URL
`github`	Your GitHub URL
`weight`	Lower = appears first

`github.com/LiuzLab/Liuzlab.github.io/blob/main/`

`src/content/team/ap-hyun-hwan-jeong.md`

Step 3 (cont.): The Markdown Body

Below the -- frontmatter, write your extended bio in **Markdown**:

```
---
... (frontmatter above) ...
---
Dr. Jeong is an Assistant Professor specializing in
computational biology and bioinformatics.

His notable contributions include **SalmonTE**, an ultra-fast
pipeline for quantifying transposable elements from RNA-seq
data, and **CRISPRcloud**, a cloud-based platform for CRISPR
pooled screen analysis.
```

Tips for the body text

- Keep it to **2–3 short paragraphs**
- Use `**bold**` for key tools or project names
- This appears on your **full profile page**, not the team card

Profile Fields Reference

Field	Required?	Description
`name`	Yes	Your full name
`role`	Yes	One of: *PhD Student*, *Postdoc*, *Master Student*, etc.
`avatar`	Yes	Path to your photo in `src/assets/`
`title`	No	Academic titles (array)
`bio`	No	One-line description for the card
`email`	No	Your email address
`website`	No	Personal website URL
`linkedin`	No	LinkedIn profile URL
`github`	No	GitHub profile URL
`twitter`	No	Twitter/X URL
`googleScholar`	No	Google Scholar profile URL
`weight`	No	Display order (lower = first, default 100)

Valid `role` values: *Principal Investigator, Professor, Associate Professor, Assistant Professor, Postdoc, Research Assistant, PhD Student, Master Student, Undergraduate, Alumni*

Step 4: Add Your Photo

1. Prepare a **square (1:1)** photo
2. Name it descriptively: `jane-smith.jpg`
3. Place it in the `src/assets/` directory

```
# Copy your photo into the assets folder
cp ~/Photos/my-headshot.jpg src/assets/jane-smith.jpg
```

Then reference it in your profile:

```
avatar: "../../assets/jane-smith.jpg"
```

Photo tips

Professional headshot preferred. Supported formats: `.jpg`, `.png`, `.webp`

Step 5: Stage and Commit Your Changes

```
# Check what files you changed
git status

# Stage your profile and photo
git add src/content/team/phd-jane-smith.md
git add src/assets/jane-smith.jpg

# Commit with a descriptive message
git commit -m "Add Jane Smith member profile"
```

Commit message tips

- Use imperative mood: “Add”, “Update”, “Fix” (not “Added”)
- Be specific: “Update Jane’s bio and photo” not “Update stuff”

Step 6: Push to Your Fork

```
# Push your branch to your fork (origin)
git push origin update/your-name-info
```

This uploads your branch to **your fork** on GitHub — it does **not** affect the main lab repository yet.

Step 7: Create a Pull Request

```
# Create a PR from the command line
gh pr create \
  --title "Add Jane Smith member profile" \
  --body "Added my profile and headshot photo.

## Changes
- New file: src/content/team/phd-jane-smith.md
- New photo: src/assets/jane-smith.jpg

## Preview
Please preview the /team page after merging." \
  --repo LiuzLab/Liuzlab.github.io
```

The `gh pr create` command:

- Opens a PR from **your fork** → **upstream main**
- Adds a title and description
- Returns a **URL** you can share

Step 8 (Optional): Test Locally

Want to preview the website before submitting a PR?

```
# Install dependencies (first time only)
npm install

# Start the local dev server
npm run dev
```

Visit `http://localhost:4321/team` in your browser to see your profile.

Requirements

Node.js 18+ and npm installed. Check with `node --version`

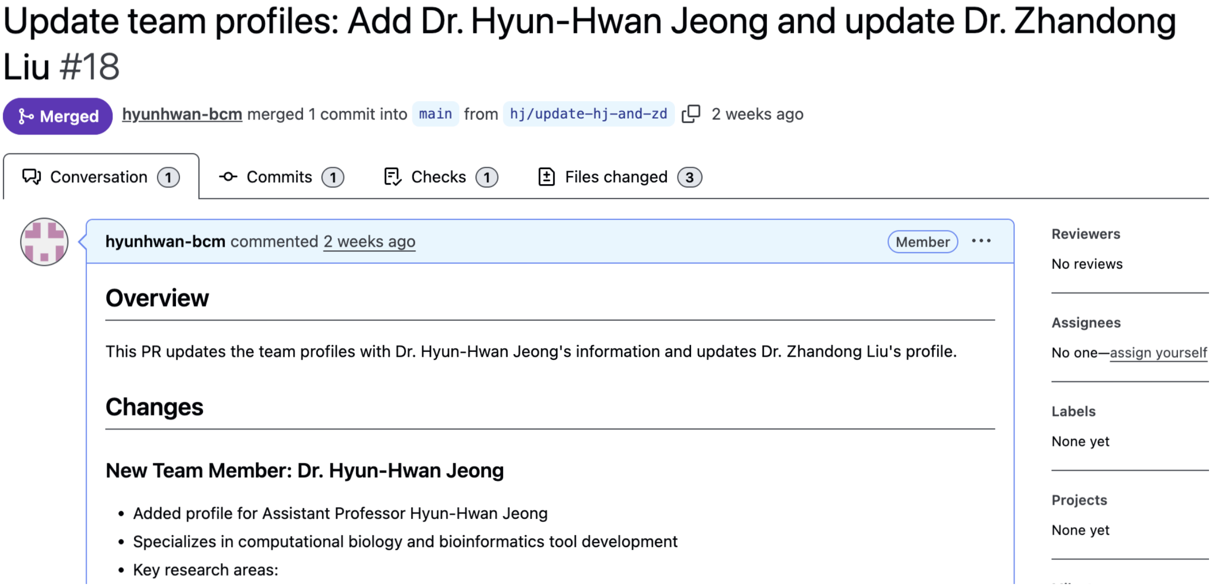
Checking PR Status and Review

```
# View your PR in the browser
gh pr view --web

# Check the status of CI checks
gh pr checks

# List all open PRs on the repo
gh pr list --repo LiuzLab/Liuzlab.github.io
```

What happens next:

1. A reviewer (maintainer) looks at your changes
2. They may request changes or approve
3. Once approved, the PR is **merged** into `main`
4. The website auto-deploys via GitHub Pages

What Your PR Should Look Like



Update team profiles: Add Dr. Hyun-Hwan Jeong and update Dr. Zhandong Liu #18

hyunhwan-bcm merged 1 commit into `main` from `hj/update-hj-and-zd` 2 weeks ago

hyunhwan-bcm commented 2 weeks ago

Overview

This PR updates the team profiles with Dr. Hyun-Hwan Jeong's information and updates Dr. Zhandong Liu's profile.

Changes

New Team Member: Dr. Hyun-Hwan Jeong

- Added profile for Assistant Professor Hyun-Hwan Jeong
- Specializes in computational biology and bioinformatics tool development
- Key research areas:

Reviewers
No reviews

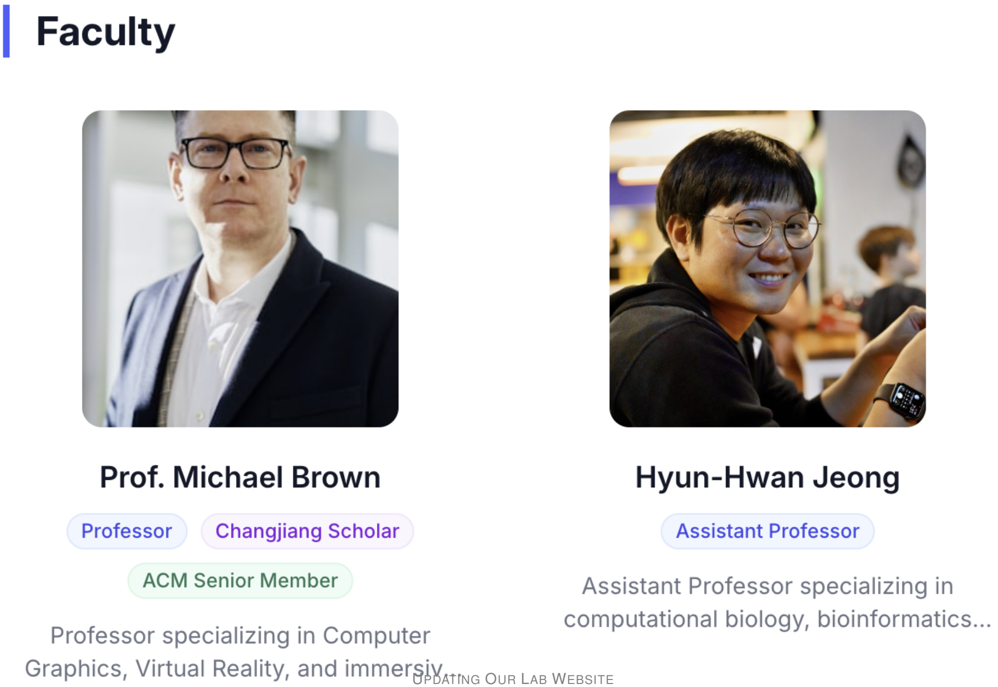
Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Your PR should include a clear title, description, and show the files you changed.

Expected Output: Team Page



Faculty

Prof. Michael Brown

Professor · Changjiang Scholar · ACM Senior Member

Professor specializing in Computer Graphics, Virtual Reality, and immersiv...

Hyun-Hwan Jeong

Assistant Professor

Assistant Professor specializing in computational biology, bioinformatics...

Quick Reference: Complete Workflow

```
# 1. Fork & clone (one-time setup)
gh repo fork LiuzLab/Liuzlab.github.io --clone
cd Liuzlab.github.io

# 2. Sync and branch
git checkout main && git pull upstream main
git checkout -b update/your-name-info

# 3. Make your changes
cp ~/my-photo.jpg src/assets/your-name.jpg
# Edit src/content/team/role-your-name.md

# 4. Stage, commit, push
git add src/content/team/role-your-name.md
git add src/assets/your-name.jpg
git commit -m "Add your-name member profile"
git push origin update/your-name-info

# 5. Open a pull request
gh pr create --title "Add your-name profile" \
  --body "Added my profile." \
  --repo LiuzLab/Liuzlab.github.io
```

SUMMARY

1. **How:** Fork → Branch → Edit → Commit → Push → PR
2. **Why:**
   1. **Decentralize** — You own your profile. No bottleneck.
   2. **Learn GitHub** — Same workflow used everywhere in software and research.

Your action item

Update your profile on `LiuzLab/Liuzlab.github.io` and submit a PR!

Stuck? Ask AI for Help!

If you encounter any Git/GitHub issues:

1. **Ask an AI assistant:**
   - Claude, GitHub Copilot, or Gemini
   - Describe your error message or problem
   - Get step-by-step solutions
2. **Common issues they can help with:**
   - Merge conflicts — “How do I resolve this conflict?”
   - Push rejected — “What does ’non-fast-forward’ mean?”
   - Commit mistakes — “How do I undo my last commit?”
   - Branch issues — “How do I switch branches?”
   - Workflow guidance — “Put all changes to a new branch, then make PR”

Pro Tip

Copy-paste error messages to AI. It will explain what happened and suggest fixes (force push, rebase, reset, etc.)

Troubleshooting

Questions?

`github.com/LiuzLab/Liuzlab.github.io`

liuzlab.github.io