

MorphoDepot

GitHub for Segmentation Projects

Big Picture Goals

- Faciliate open science analysis of biological specimens
- Tools and methods for scalable collaboration
- Simplify contributions for anyone (not just lab members)
- Make system as friendly as possible for new users
- Track user contributions (e.g. for co-authorship)
- Ideally usable for many purposes, but SlicerMorph users are first priority

Advantages of GitHub

- Free/inexpensive, widely used, mature
- Traceable, open science
- Already used in MorphoCloud

Use case summary

- Specimen scans that need annotation (e.g. segmentation or landmarks)
- A team of contributors
- A project leader to organize and approve contributions

For example:

- microCT scan of a frog species
- PI interested in anatomical segmentation
- several students in a class assigned to segment different parts (limbs, body, head...)

How this maps to GitHub

- PI and students all create accounts on github.com (free tier is okay, just need email address)
- Segmentation project is stored in a repository
- Scan can be stored at any URL (e.g. s3 or JS2 bucket, or as github asset) in standard formats (nrrd, zarr...)
- Repository follows MorphoDepot template (source URL, anatomy terminology, resulting segmentation)
- Students are assigned issues for segmentation tasks
- Students create pull requests for their assigned issues
- PI reviews and merges pull requests

Other nice features of GitHub for MorphoDepot

- We can use the GitHub API to automate and simplify many steps
- GitHub handles user accounts, email integration, sys admin, and many other complex issues
- All GitHub project management features available to organize work
- GitHub repository "topics" can be used to organize, e.g. by species or developmental stage
- Readme files can include links, images, experiment descriptions, etc.

Big Picture Issues

- GitHub is designed for code, not 3D volumes
 - so text-based tools may not be useful
 - we may need to create custom data-specific replacements
- GitHub is built for code developers, not biologists
 - so tools can be difficult to use
 - we may need to develop automation or custom documentation
- GitHub is a commercial service and terms may change at any time
 - it's been around since 2008, owned by Microsoft since 2018, so we can hope it's stable
 - there are some alternatives if needed someday (GitLab,

3D Slicer Integration Plans

- A MorphoDepot module in SlicerMorph to simplify the workflow for contributors
 - List issues assigned to the contributor (or issues available to claim)
 - User can select an issue to work on
 - Automatically creates branches and loads the data
 - Allows user to save work in progress to pull request
 - User can mark a contribution as ready for review
- MorphoDepot can be used on a local machine or a Jetstream2 virtual desktop

MorphoDepot Status

- Initial prototype under development
- Feedback from MorphoCloud user community wanted!
 - pre-alpha versions could be ready for testing
 - need segmentation projects with concrete use cases to work out details

Discussion Points: Authentication model

- Ensure integrity of repository
 - Accurately track individual contributions
 - Don't allow users to delete or overwrite other contributions
- Respect user accounts
 - Don't give TA or fellow students access to others' accounts
 - Don't expose high-value credentials on shared machines
 - Encourage users to rely on GitHub for their other projects
- Streamline login for new users
 - Simplify/automate but don't obscure underlying security model
 - Use existing GitHub tools when possible

Discussion Points: Repository architecture

- Single specimen / scan or multiples
- Structured metadata about specimen
 - Use of GitHub topics to find repositories
 - JSON Schema for describing segmentation tasks
- Best storage for original scans (GitHub assets, JS2 buckets, etc.)
- Fork vs branch model / are contributors part of the repo