生物物理若手の会 北海道支部会

生命科学者のための、Git/Githubハンズオン

物理工ソロジー研究室 D1 髙橋奏太

自己紹介・ITにつよつよ風

- 髙橋奏太(たかはし そうた)
- 北大生命科学院ソフトマター専攻物理エソロジー研究室 D1
- 兵庫県出身
- 2017年に北大理学部生物科学科高分子専修に配属
- 2018年から2年休学して
  - 株式会社スマートルアー共同創業(IoTベンチャー。2年で消滅。調達n億/売上0円の典型的(以下略))
  - 自社メディアの制作・運営やUXデザイン。
  - やめて個人事業:小売しつつECサイト開発/レビューサイト開発/AR開発とか

1. Git とは?

- 文字で書いてあるものになら何でも使える「**セーブポイント**」システム。
  - 本質は「差分の管理のためのシステム」といえる。
- 「あのときの変更」を、いつでも「あのとき」に戻せる。
- 「あのとき」の変更を、いつでも「今」に戻せる。
- **なんなら複数ルートを同時に進められる**

👉 **「編集しているうちに誤操作でうっかりバグが入った」「2023_09_21_髙橋最終版_変更済み_v2」「コメントだらけで逆にわけわからない」「本番環境からコピペして、手元で編集して、本番環境にコピペして…」** といった問題を解決できる。

- 解析コード、計算コードのデバッグの強い味方。変更が怖くなくなる!
- 論文執筆に使えばブラッシュアップの過程が一目瞭然!

Github は Git管理されているフォルダをオンラインで管理&共有するサービス

つまり...

- 他人のセーブポイントにジャンプできる!
- 他人のルートに口出しできる!(言い方が悪い)
- 自分の別デバイスでの同期にも使いやすい!

👉 共同作業が爆速に

- Word の「校閲」機能の上位互換のようなもの。
  - 二重三重のやり取りもお手の物です。
- 「botとの共同作業」も可能。
  - 例えば、文献管理ソフトとの連携が強力。

本筋とは関係ないけど、なぜか使える生成AIのGithub Copilotはすごい(GPT-4/無料)。

両方強力だけど最初の「黒い画面」は避けがたい…💦

みんなでやろう✨

Gitを始めてみよう:Mac編-1

1. ⌘+Space でSpotlight検索を開き、「ターミナル」と入力してターミナルを開く。
   「黒い画面」...!
2. `git --version` と入力して、バージョンが表示されればOK。
   - 表示されない場合は、こちらからインストール。

Gitを始めてみよう:Windows編-1

1. Git for Windows をダウンロードしてインストールする。
2. スタートメニューから「Git Bash」を起動する。
   「黒い画面」...!
3. `git --version` と入力して、バージョンが表示されればOK。

Gitを始めてみよう:Mac編-2

1. 作業用のフォルダを作成する。
    - 例: `mkdir ~/GitHub_test`
2. 作業用のフォルダに移動する。
    - 例: `cd ~/GitHub_test`
3. `git init` と入力して、初期化する。
    - 「Initialized empty Git repository in ～」と表示されればOK。

Gitを始めてみよう:Windows編-2

1. 作業用のフォルダを作成する。
    - 例: `mkdir C:\Users\username\GitHub_test`
2. 作業用のフォルダに移動する。
    - 例: `cd C:\Users\username\GitHub_test`
3. `git init` と入力して、初期化する。
    - 「Initialized empty Git repository in ～」と表示されればOK。

Gitを始めてみよう:Mac編-3

1. 作業用のフォルダにファイルを作成する。
    - 例: `touch README.md`
2. `README.md` にこのフォルダの説明を書く。
    - 例: `echo "# GitHub_test" >> README.md`
3. `git add README.md` と入力して、ファイルをセーブポイントに追加する**準備**。
4. `git commit -m "first commit"` と入力して、ファイルをセーブポイントに追加する**実行**。
    - 「1 file changed, 1 insertion(+)」などと表示されればOK。

Gitを始めてみよう:Windows編-3

1. 作業用のフォルダにファイルを作成し、このフォルダの説明を書く。
    - 例: `echo "# GitHub_test" >> README.md` ((内容)>>ファイル名)
2. `git add README.md` と入力して、ファイルをセーブポイントに追加する**準備**。
3. `git commit -m "first commit"` と入力して、ファイルをセーブポイントに追加する**実行**。
    - 「1 file changed, 1 insertion(+)」などと表示されればOK。

Gitで遊んでみよう:1

1. `README.md`を編集する。
    - なんでもいいから追記する。おすすめは「なかったことにしたいハプニング」
    - コマンドプロンプトをつかった例: `echo "Hello World!" >> README.md`
    - メモ帳、テキストエディットなども全然OK。
2. `git add README.md` と入力して、ファイルをセーブポイントに追加する**準備**。
    - 使ったエディタによっては保存してからこの操作を行う必要があるかも
3. `git commit -m "具体的な編集内容"` と入力して、ファイルをセーブポイントに追加。
    i. `git commit -am "具体的な編集内容"` と入力して、準備と実行を同時にすることもできる。
    ii. この編集内容のメモを「コミットメッセージ」という!そのセーブポイントがどんなものか後で知るために最重要なので、`first commit`以降はできるだけ丁寧に書こう。

Gitで遊んで見よう:2

1. `git log` と入力して、セーブポイントの履歴を見る。
    - 「commit ～」(～はデタラメな英数字)というのがセーブポイントのID。コミットメッセージをみて、飛びたいセーブポイントを見つける。
2. `git checkout ～` と入力して、セーブポイントにジャンプする。
    - 「HEAD is now at ～」と表示されればOK。
3. `README.md` を見てみると、「なかったことにしたいハプニング」が消えていることがわかる。

Githubに登録してみよう

1. Github にアクセスして、アカウントを作成する。
2. 「Create a repository」をクリックして、リポジトリを作成する。
    - 「Repository name」には、先ほど作成したフォルダ名を入力する。
    - 「Initialize this repository with」には、先ほど作成したフォルダの中身を入れる。
    - 「Create repository」をクリックして、リポジトリを作成する。
3. 「Quick setup」の「...or push an existing repository from the command line」のコマンドをコピーする。
    - 例: `git remote add origin`
4. `git` のユーザー名とメールアドレスを設定する。
    - 例: `git config --global user.name "Kota Takahashi"`
    - 例: `git config --global user.email "hoge@hoge.com"`

そろそろ黒い画面に疲れてきたので、VSCodeも使ってみよう

安心のMicrosoft製エディタ。Githubとの親和性が高い。

1. VSCode をダウンロードしてインストールする。
2. 立ち上げて「フォルダを開く」または「リポジトリを複製」で先程のフォルダを開く。
3. 「Source Control」タブを開けば、`git` コマンドの操作は全部ここからできます。

総まとめ: Githubで共同作業をしてみよう

Special Thanks : 参加者の方々

おまけ: 研究生活におけるGitHubの活用例

1. 論文執筆

一つの作業用リポジトリを

- 参考文献管理の全自動化
- 論文の原稿のバージョン管理

に利用。DXの申請書が一枠これで埋まりました笑
差分比較機能をつかって、先生がどこを変えたのか、なぜだめだったのか、を振り返って圧倒的成長 💪

2. 解析・可視化用コードの管理

とくに、

- 解析手法をちょっと変えてうまくいくかわからないとき
- 同じデータをバージョン違いで可視化するとき
- 家や出先など複数PCから作業するとき

に便利だった。
ファイルを増やしてバージョン管理をやろうとすると際限なく増えてどれがどれだか分からなくなる。
差分ごとに同期ができるので、どこで作業しても最新の状態で作業できる。Google Drive等だといちいち丸ごとコピーする必要がある。

3. github pagesでのウェブサイト作成

4. 草が生える

進捗が可視化される気の利いた機能。
研究生活のささやかな精神補助。