# 数理生物学演習 第1回 Hello, World!

## 野下浩司(Noshita, Koji)

☑ noshita@morphometrics.jp 命 https://koji.noshita.net

理学研究院 数理生物学研究室





## 野下浩司(Noshita, Koji)

☑ <u>noshita@morphometrics.jp</u> <u>https://koji.noshita.net</u> 九州大学 理学研究院 生物科学部門 数理生物学研究室 九州大学:植物フロンティアセンター、マス・フォア・イノベーション卓越大学院、名古屋大学:生命理学専攻 |異分野融合生物学講座,奈良文化財研究所 埋蔵文化財センター 遺跡・調査技術研究室

これまでの研究:数理・情報科学を駆使した生物の**形態進化を解明する研究と他分野への応用** かたちのモデル化・定量化 生物の形態進化の階層性・ 3D形態情報の取得と解析法の開発 適応地形 動物 適 応 植牧 度 形態空間 人工物 発生モデル 遺伝子型と表現型の関係性 ゲノム予測 **GWAS** 遺伝子空間 1:0.672 Coeff: 0.7711 点群データに基づく評価 農学への応用 医学への応用 マクロファージの分布推定 ۰ 葉面積 草丈 葉数 など 葉形状 曲率分布 など



## 生物や生物のつくる構造物の 「かたち」を研究しています

## 形態進化を 究極要因と 至近要因の両面から 定量的に理解したい

キーワード:形態測定学,理 論形態学, 画像解析, 機械学 習、植物フェノタイピング、 貝類学,考古学,





相撲

観戦















## コンピューターによる数値解

 $\mathbf{x}(t)' = -\mathbf{d}\mathbf{x}(t) - \mathbf{\beta}\mathbf{x}(t)\mathbf{v}(t), \ \mathbf{y}(t)' = \mathbf{\beta}\mathbf{x}(t)\mathbf{v}(t) - \mathbf{a}\mathbf{y}(t), \ \mathbf{v}(t)' = \mathbf{k}\mathbf{y}(t) - \mathbf{r}\mathbf{v}(t)$ 





数式による定式化



 $\frac{dr}{ds}$  $= E_{\rm G}(s)r(s)$  $egin{array}{c} \mathcal{C}(s) \ 0 \ -\mathcal{T}(s) \end{array}$  $0 \ \mathcal{T}(s)$  $\frac{d}{ds}$  $-\mathcal{C}(s)$  $\xi_2$ =

 $\xi_1$  $\xi_2$  $\xi_3$ 

## 複雑な現象のシミュレーション





データ解析



## コンピューターを使った解析はあらゆる領域で一般的な方法になった. 今後もますますこの傾向は強くなる。"AI"ツールの活用も急速に進む。







本演習の進め方

演習の基本方針

★★★ 数理生物学的トピック:数理生物学に興味を持ってもらう 例: ロトカ-ボルテラモデルによる被食者-捕食者の個体群動態の記述

★★ 研究への挑戦:研究を進める上で必要な、または役立つ経験をする 例:文献の探し方、使うと便利なツールの紹介

★ 数値計算的アルゴリズム:計算機で問題を解く方法を学ぶ 例:ニュートン法により方程式を数値的に解く

★★★ プログラミングのスキル:実際に具体的な問題を解いてみる

例:if文、forループ、配列

# 自分の知りたいことへアプローチする方法を学ぶ

好きなタイミングで好きなだけ取り組む。締め切りまでに課題を提出すればOK.

第n-1回 トラブルシュー ティング 解説動画の視聴 • 解説スライド ノーマル課題へ の取り組み  $+\alpha$  ハード課題へ の取り組み 内容や課題に関する質問は 随時Slackなどで受け付ける

演習の進め方:基本的な取り組み



### 変更があればWebページにて連絡.

今後の予定

- 1. Hello, World!
- 2. Pythonの基本的な使い方
- 3. 個体群動態の数理モデル(1):離散ロジスティック成長モデル 4. 個体群動態の数理モデル(2):指数増殖モデル、ロジスティック成長モデル 5. 個体群動態の数理モデル(3):ロトカ-ヴォルテラ モデル
- 6. ランダムな現象:突然変異固定までの待ち時間
- 7. 空間構造の数理モデル(1):理論形態学, Raupのモデル
- 8. 数理生物学は役に立つのか?(1):研究紹介
- 9. 研究をはじめるために:論文の検索,管理,引用 10.空間構造の数理モデル(2):人工生命、セルオートマトン、(ライフゲーム)
- 11.空間構造の数理モデル(3):パターン形成. (拡散反応モデル)
- 12.数理生物学は役に立つのか? (2):研究紹介
- 13.数理生物学は役に立つのか? (3):研究紹介
- 14.数理生物学でのプログラミング:まとめ、最終課題へ向けた調整

Webページ(後ほど紹介)を随時確認すること!

- 現時点での予定なので、今後変更される可能性あり、

- ・ 課題提出:基本毎回提出. 出席を兼ねる. GiHubにて提出.
  - ノーマル:操作説明と内容説明を視聴して,ある程度の 試行錯誤と調べ物を行えばできるであろう応用的な課題

  - ハードは希望者のみ(取り組みの量や難易度に成績が比例しない).
  - ノーマル:60点(平均目安:50点)
  - ハード:15点
- 最終課題:GitHubにて提出。

  - ノーマル:35点(平均目安:25点)
- その他:5点
- ・ ボーナス **バグ報告**:Slackの「#バグ報告」チャンネルで報告.
  - ・ 単純なバグ指摘 ~ 修正の提案などに応じて1つの報告につき最大2点
  - 同じバグについて基本は先着1名. 誰かが報告したバグに修正案を出してもOK.



GPA目安

- S Excellent 90以上
- A Good 89-80
- B Satisfactory 79-70
- C Pass 69-60
- F Fail59以下

理論的最大点

• 120

• ハード:自分で積極的に調べたり試行錯誤を繰り返したりしなければできないであろう発展的な課題

## 数理生物学的なテーマを設定し、計算機を使ったアプローチで取り組み、レポートにまとめる













### 手を挙げて or Slackなどで積極的に質問をしよう。きっとTAが助けてくれる。











れば、より知りたいと思った点を教えて、

2.今後扱って欲しい内容を挙げる(複数可)

3. 「Hello, World!」 プログラムを作成せよ.

ハマってます、など)

5.その他質問、感想、要望をどうぞ、

本日の課題 ノーマル

- 1.今回の授業でよくわからなかった点を述べる. 特になけ
- 4.Slackの「#自己紹介」チャンネルに自分の自己紹介を投 稿する、(例、~~です、趣味は~~です、最近~~に

- !注意:第1回課題提出の締め切りは、第2回の前日23:59まで.
  - 課題をGitHub Classroomにて提出すること



- Webページ
   <u>https://koji.noshita.net/ce</u>
- email 野下浩司 <u>noshita@morphometrics.jp</u>
- Slack(後ほど説明します)
   <u>ku-compbio-2025.slack.com</u>
- 訪問(事前連絡してください)
  - 伊都キャンパス ウエスト1号館 W1-C-906
- 研究室の様子を覗きに来るのも可 その場合はW1-C-909付近で誰か見つけて話しかけて!

## https://koji.noshita.net/courses/compbio/compbio2025/







# 第1回:Hello,World!

## 本日の目標

本演習で使うプログラミング言語の紹介

# プログラミング言語って?



## このままでは分かり合えない → 互いの言語の"翻訳"が必要







"翻訳":コンピュータ → 人





	ブ言語	しいえ	ろし	۱3	
発 <b>C</b>	Go Ty	新進気鋭 peScrip	ot		
	Rust Elix		Julia		
	数式処理		Perl		
a I	Magma Wolfram	PHP	Javas	script	
OCa k omn	aml Scala non Lisp	Web	開発	Ruby	
関数 Ma	女型 tlab L	ua	R デ・	<mark>Pythc</mark> ーク解析	<b>on</b>
50	٦٢ ٢				
<b>γ γ</b> -	- <b>フ</b> い ス の ズ F	中中一一		イン: 動的 扱い	- タプリタ /型付け /やすい



# Puthon Compared Compared

# インタプリタ型、動的型付け

## Python

https://www.python.org/

用途:数値シミュレーション、可視化、データ解析

Pythonは数理生物学的なシミュレーションだけなく, データ 解析や画像解析などにも利用されている → 生物の分野で扱うデータやその解析にも利用できる. 様々 な環境(ラボとか職場とか個人開発とか)へ応用しよう。

## 特に理由は無いけれどプログラミング言語を 学ぶと最初に出てくる練習用プログラム

## 目的 「Hello, World!」と出力(画面に表示)する

- きちんと環境の設定を出来ているか
  - Pythonはインストールできている?
  - パスは通っている?
  - 開発環境の準備は?

Hello, World!

## …などをチェックする.

# プログラミング環境を準備しよう!

## https://koji.noshita.net/courses/compbio/compbio2025/

本演習の情報は基本的にここに公開される。ブックマーク推奨。

演習 Webページ

https://moodle.s.kyushu-u.ac.jp/

moode

## 演習での活用方法

- ・最初の情報のやり取り
- ・学籍番号と紐づく情報の確認

## 登録方法

- 1. 九州大学Moodleへログイン(SSO-KIDとパスワードが必要)
- 2. コースを検索する → 「数理生物学演習」をキーワードに検索
- 3. 「2025年度前期・月3月4・数理生物学演習」を選択
- 4. 「私を登録する」をクリック

# 九州大学Moodle

オープンソースeラーニングプラットフォーム「Moodle」を利用した 九州大学のオンライン学習システム(の一部)



## **slack** チャット系のコミュニケーションツール

### 演習での活用方法

- ・質問する
   「#質問」チャンネル
  - 例)インストールが~で止まる、~の図がきれいに表示されない。
- ・他の人の質問に答える 「#質問」チャンネル
  - 例) ~にチェックを入れるといいよ、途中の~の計算ができてないん じゃない?
- ・情報を共有する 「#general」「#雑談」チャンネル
  - 例) ~の参考になるページを見つけた、~は~からダウンロードできる
- ・ バグを報告する 「#バグ報告」チャンネル ボーナス点あり
  - コードの誤りやスライド・資料の間違いを指摘する
  - 例) ~を実行しても~にならない。~ではなく~ではないか?
- その他にも趣味の話など様々に活用してください。

### 登録・利用方法

- PCやスマートフォン用のアプリがあるのでインストールすることを薦める。

個人情報を含むやり取りなどの場合のみ利用可 |**#質問**チャンネルを活用してください. わからないことや曖昧な点をオープンな場で質 |問・議論し、理解を進めるという経験を積んで| ほしいです

・大学メールアカウント(~@s.kyushu-u.ac.jp)へ招待を送るので,そこからアカウントの作成をおこなう



- それに加えて、スマホのアプリなどのインストールも推奨



### 好きな方から導入してください

Slack デスクトップアプリの導入

## • Slackへはブラウザや各プラットフォーム向けのアプリからアクセスできる ・ 演習では **演習に用いるPCにデスクトップアプリをインストール**してください

### ダウンロードページ (<u>https://slack.com/intl/ja-jp/download</u>)



### 好きな方から導入してください

- 氏名(full name)の設定
  - 漢字、ひらがな、アルファベットのいずれでも構わないが、本名で設定すること
  - 例:野下 浩司,のした こうじ,Koji Noshita
- 表示名(dispaly name)の設定
  - ・名字\_学籍番号(名字はアルファベット、すべて半角) ↑アンダーバ-
  - 例:noshita\_1SC20\*\*\*\*
- ・プロフィール写真の設定(次回までに)
  - 絵でも, 写真でも, 文字でも良いが, 他と見分けやすく, 混乱が生じにくいものを作 成・採用する.
  - 目安
    - 画像サイズ 500px x 500 px 以上(四角形がおすすめ)
    - ファイルサイズ 1MB以下
    - (透明部分は無いほうが良い) ・フォーマット png

Slack登録後にやること

氏名		プロ
野下 浩司		
表示名		
noshita		
これは名字や名前、ニック メンバーから呼ばれたい4	クネームなど、好きに設定できます 名前にしましょう。	ි Slack で
役職・担当		ī
役職・担当		
数理生物学演習2020年度	でのあなたの役割を説明しましょ	〕 う。
電話番号		
(123) 555-5555		
電話番号を入力してくださ	さい。	)
タイムゾーン		
(UTC+09:00) 大阪	反、札幌、東京	~
、 現在のタイムゾーン。サマ ィードやリマインダーのB	マリーや通知メールの送信、アクテ 時間に使われます。	, ィビティフ



## Google アカウント https://myaccount.google.com

# Google

## Google アカウントの作り方 (**既に持ってる人は新たに作る必要はありません**」それを使ってください)

Googleアカウントのページ(<u>https://myaccount.google.com/</u>) ヘアクセス

Google アカウント 概要 プライバシーツール

- 「Googleアカウントの作成」をクリック
- アカウント情報を入力
  - 名前、ユーザー名(これがGmailのアドレ) スになる)、パスワード
- 電話番号を入力
  - SMSで確認コードが送られてくるので、 それを入力

Googleに関連するサービスを利用するためのアカウント. 既にGmailやGoogleドライブを利用している人は,利用しているアカウントがこれに当たる.



- ・生年月日、性別を入力
  - 再設定用メールアドレスを入力しておくとパス

ワードを忘れた際に助かる

- プライバシーポリシーへ同意
  - 問題がなければ同意する







Googleのオンラインストレージサービス 本演習では、コードやその実行結果などを保存する. また、Colab(後述)の外部ストレージとして利用する.

## 演習での活用方法

- ・コード等の保存
  - サンプルコード、自分が作ったコード
  - ・出力した図、ファイル など
- Colabの外部ストレージ
  - Colabにマウントして、結果の出力

最低150MB以上の空き容量を確保しておくこと. もし不足している場合は、必要のないファイルの削除等の対応をしてください。

Google ドライブ

https://www.google.com/drive/

Colabに読み込ませるデータをGoogleドライブにアップロード など

### Googleドライブ内に演習用フォルダを作る 1. 「マイドライブ」の直下に「CompBio2025」というフォルダを作る 🔼 ドライブ **Q** ドライブで検索 ? $\oslash$ HH 2. 「新規」から新しいフォルダを作成 新規 名前 个 オーナー 最終更新 ☑ マイドライブ $\bigcirc$ 候補 - [ ] **\*\*\*\*** 「マイドライブ」を選択 マイドライブ 10-10-10-10 (B19) 履歴 詳細 共有ドライブ Colab Renderske 3 共有アイテム 最近使用したアイテム 回りたりたらる 御道 (game, Should \$3 スター付き infographics 調査を行ると言い意思が ます ゴミ箱

## 2. 「CompBio2025」の中に、「01」というフォルダを作る

マイドライブ > CompBio2020 -

名前个

01

Sectors.

## 今回は、この「01」の中にコードを配置していきます.

次回以降は新たに02,03,…と回数を名前にしたフォルダを「CompBio2025」の直下に作って, 毎回その中にコードを配置するようにしてください



オーナー	最終更新
自分	22:33 自分

# Gitレポジトリホスティングサービス,開発プラットフォーム. 本演習では、主に課題の提出やフィードバックに利用する.

## 演習での活用方法

- 課題の提出
  - GitHub Classroomを介した課題の提出
- 課題へのフィードバック

  - ・課題のわからない部分などを質問・議論
- 各自のコード管理

# GitHub

https://github.com/

教員やTAがPull Requestなどを利用してフィードバック

演習に用いたコードの管理などに自由に利用可能

### GitHubアカウントの作成 1. サインアップ Product V Solutions V Resources V Open Source V Enterprise V Pricing Search or jump to. Build and ship software on a single, collaborative platform Join the world's most widely adopted Al-powered developer platform. Try GitHub Copilot Enter your email Sign up for GitHub Sign up to GitHub Email ----Email Password メールアドレス,パスワード、ユーザ Password Password should be at least 15 characters OR at least 8 characters including a number and a lowercase lette -名などを入力し、アカウントを作成 Username Username Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen. Your country Japan アカウント作成の際に、非ロボットの For compliance reasons, we're required to collect country information to send you occasional updates and announcements. 確認、メールアドレスの確認、などが **Email preferences** Receive occasional product updates and announcements ある場合には対応する. Continue >

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.





## 2.サインイン

https://github.com/login



サインアップが完了 後にジャンプした場 合は、以下のような 表示がある。

Your account was created successfully. Please sign in to continue

作成したアカウント情報でログイン





# GitHubアカウント作成後にやること

- ユーザー名 (username) の設定
  - アカウント作成時に設定している。必要があれば、変更可能。
  - 本名である必要はない.
- 表示名(name)の設定
  - 必要に応じて変更する.
  - 本名である必要はない.
- ・メールアドレス
  - アカウント作成時に設定している。必要があれば、変更可能。
- プロフィール写真の設定(次回までに)
  - 絵でも,写真でも,文字でも良いが,他と見分けやすく,混乱が生じ にくいものを作成・採用する.
  - 目安
    - ・
       ・
       画像サイズ
       500px x 500 px 以上
    - ファイルサイズ 1MB以下
    - フォーマット png (透明部分は無いほうが良い)
  - Slackのものと同じでも、異なっていても構わない。

A Public profile	Change username
龄 Account	
& Appearance	Changing your username can have <u>unintended side effects</u> .
者 Accessibility	Change username
Notifications	② Looking manage account security settings? You can find them in the Password and au



Name	Profile pictur
Koji Noshita 😶	
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.	
Public email	
noshita@morphometrics.jp 🗢 🗙 Remove	
You can manage verified email addresses in your email settings	
Bio	C Edit
Tell us a little bit about yourself	

## Slackとは異なり世界に公開され る情報です、注意してください。







# GitHubアカウント (username) の提出

1. Moodleの「GitHubアカウン	ント名の提出」
2025年度前期・月3月4・数理生物学演	<sub>賢習</sub> (野下 浩司)
コース 設定 参加者 評定 レポート さらに →	
~ 一般	すべてを折りたたむ
GitHubアカウント名の提出	
GitHubアカウントを作成し,アカウント名を提出してください.	

## 3.GitHubアカウント名を入力して提出



## 課題提出に使うので必ず提出してください

## 2.アンケートに回答する



## GitHubのプロフィールページ

https://github.com/XXX

XXXがアカウント名



# 皆さんへのお願い

- わからないところがあればすかさずググろう!
   調べる習慣をつける.
- ・ 質問や回答をSlackへ投稿しよう. 情報が共有できる. 一人の質問が皆の質問に!
- 困ったら(手を挙げて/Slack上で)助けを呼ぼう(特に、TAがサポートしてくれる).困っている人がいれば助けてあげよう.
- ・演習中の休憩は自由、疲れ果てる前に休もう、

# ColabによるPythonの実行

- 対話型評価環境 る。ちょっとした計算や確認に用いられる。
- ・ソースコード るコード(パッケージ)を利用する場合に用いられることが多い.

ノートブック セルと呼ばれる入力・出力エリアからなる実行環境であり、コードやその出 力結果、注釈などのテキストを記録できるファイル、データ解析を行う際の 環境としてよく利用される。

Pythonの実行方法いろいろ

式やコードを入力し、それをインタープリタが解釈・実行し、結果を表示す

プログラミング言語で書かれたテキストファイル(=ソースコード)を読み 込んで、上から順に解釈・実行していく、大規模な計算や繰り返し用いられ

本演習では主にColab上でノートブックを利用して進めていきます



## Googleが提供しているJupyter notebookのフォーク・計算環境 本演習では、Colabを使って、Pythonのコードを実行していく

⊨ 目	次	×	+ コード + テキスト 🛯 🍐 ドライブにこ
↔	はじめに データ サイエンス 機械学習 その他のリソース 機械学習の例 ■ セクション		COlaboratory とは Colaboratory(略称: Colab)では、ブ ・構成が不要 ・GPU への無料アクセス ・簡単に共有
		,	Colab は、学生、データ サイエンティ からすぐに使ってみることもできます ・ はじめに ご覧になっているドキュメントは静的 環境です。
			<ul> <li>にこえば入のコートとかには、直を言</li> <li>seconds_in_a_day = 24 * 60 * 60 seconds_in_a_day</li> <li>86400</li> <li>上記のセルのコードを実行するには、 カット「command+return」または「</li> <li>1つのセルで定義した変数は、後で併</li> </ul>

- クラウド(環境設定が必要ない)
- (基本的に)無料
- ・演習に必要なライブラリが(ほぼ)導入済み

# Google Colab

https://colab.research.google.com



セルで使用できます。

### Googleドライブをマウントできる

- GPU環境も利用可能
- など便利な機能満載

-2012 2013 2014

## IP[y]: \_\_\_\_ 💭 jupyter

IPython notebook (2011 Prototype)

Project Jupyter開始 IPython名はカーネルに残し Jupyter Notebookと改称

Google Colabとその周辺



Google Cloud DatalabのコアとしてJupyter採用

### 【秒速で無料GPUを使う】深層学習実践Tips on Colaboratory | Qiita https://qiita.com/tomo\_makes/items/b3c60b10f7b25a0a5935

Colabの使い方

## Colabへのアクセス https://colab.research.google.com/

CO Colaboratory へようこそ ファイル 編集 表示 挿入 ランタイム	G 共有 🗘 ログイン	
Ⅲ目次 ×	+ コード + テキスト ▲ ドライブにコピー 接続 ▼ / 編集 /	•
はじめに データサイエンス 機械学習 その他のリソース 機械学習の例 日 セクション	<ul> <li>Colaboratory とは</li> <li>Colaboratory (略称: Colab) では、ブラウザから Python を記述し実行できるほか、次の特長を備えています。</li> <li>構成が不要</li> <li>GPU への無料アクセス</li> <li>簡単に共有</li> <li>Colab は、学生、データサイエンティスト、AI リサーチャーの皆さんの作業を効率化します。詳しくは、Colab のご紹介をご覧ください。下からすぐに使ってみることもできます。</li> </ul>	
	<ul> <li>はじめに</li> <li>ご覧になっているドキュメントは静的なウェブページではなく、Colab ノートブックという、コードを記述して実行できるインタラクティブな環境です。</li> <li>たとえば次のコードセルには、値を計算して変数に保存し、結果を出力する短い Python スクリプトが含まれています。</li> </ul>	ĩ
	<pre>[] seconds_in_a_day = 24 * 60 * 60 seconds_in_a_day</pre>	
	<ul> <li>86400</li> <li>上記のセルのコードを実行するには、セルをクリックして選択し、コードの左側にある実行ボタンをクリックするか、キーボードショート カット「command+return」または「Ctrl+Enter」を使用します。コードはセルをクリックしてそのまま編集できます。</li> <li>1つのセルで定義した変数は、後で他のセルで使用できます。</li> </ul>	

1. Go	ogle 7	アカウン	トでロ	ゲイン
-------	--------	------	-----	-----

### 基本的には色々なパッケージ導入済みのPythonをクラウドで使えるJuputer Notebook

### **Colab内で新規作成**

### 1. ファイル



### 2. ノートブックを新規作成



## ノートブックを新規作成する(1)

### 3. Colabの操作画面が表示される

7,5,7,7,7,7,7,7	接続 🗸 🦯 編集
	1 V GO 🗐 🗘 📋 🗄
0 1	 ■ テキスト



セルのコードが上から順に実行される. テキストセル:文章を入力・出力するためのセル.

Colabの操作 画面の 概要

ツール ヘルプ	国 コメント	よ 共有	<b>\$</b>	
	接続		編集	
■ コード ■ テキスト	<u>^</u>	↓ © ■	* :	
ル(入力)				

• セル(cell):コードなどを入力する、あるいはその結果が出力される領域. コードセル:プログラムのコードを入力するセル、実行ボタンを押すと、対象の



# Hello, World!

### プログラミング言語を学ぶと最初に出てくる練習用プログラム

,	<ul> <li>コメントアウト(#)</li> </ul>
	Pythonでは#から文末までが
	(実行時に)無視される
rld!	• print(オブジェクト)
	オブジェクトを出力

接続 🗸 🎽 編集

## ノートブックに名前をつけて保存しよう!

ノートブックに名前をつける

1. デフォルトでは「Untitled[番号].ipynb」の ような名前がついている



2. 名前の部分をクリックすると編集できるの で、適切な名前をつける



演習でのノートブックの名前の付け方 サンプルコードには[数字2桁]-[数字2桁]の番号を割 り振っている(例.01-01).前半の2桁はノート ブック番号,後半の2桁はコード番号を想定している. ノートブックのファイル名は、ノートブック番号と同 (例.01)あるいはノートブック番号 内容(例. 01\_hello\_world<u>) とするのがおすすめ.</u> その他の名前でも構わないが、自分でどのノート ブックでどのような作業をしたかが把握できるように しておくこと.



1. Colabの「ファイル」 2. 「保存」をクリック

ファイル	編集	表示	挿入	ランタ	イム	ツール		
ノートブックをアップロード								
ノートブック名を変更								
ゴミ箱に移動								
ドライ	ブにコピ	ーを保	存					
コピーを GitHub Gist として保存								
GitHub	にコピー	-を保存	子					
保存					¥/	Ctrl+S		
保存	谷を保存	して固	定		ਸ/ ਸ/Ct	Ctrl+S rl+M S		
保存 変更内 変更履	容を保存 歴	して固	定		ਸ/ ਸ/Ct	Ctrl+S rl+M S		
保存 変更内 変更履 .ipynb <sup>;</sup>	容を保存 歴 をダウン	して固	]定 :		ቻ/Ct	Ctrl+S rl+M S		
保存 変更内 変更履 .ipynb <sup>;</sup> .py をク	容を保存 歴 をダウン ブウンロ-	:して団 ロード ード	ī定 :		ૠ/Ct	Ctrl+S rl+M S		

ノートブックの保存 ノートブックを閉じる

1. ブラウザのタブ(もしくはウィンド ウ)を閉じる

• • •	O Untitled0.ipynb - Colaboratory × +
$\leftarrow \rightarrow$	C colab.research.google.com/drive/1N0EQxLzaBngzJMcBu
C	🛆 Untitled0.ipynb 🛛 🕁
	ファイル 編集 表示 挿入 ランタイム
:=	+ コード + テキスト
<>	0 1

Google ドライブ内でのファイルの移動

## 右クリックから指定の場所へ移動 1. 移動させたいファイルを右クリック



## 2. 「指定の場所へ移動」

3. 移動先を選び「ここに移動」



### 演習用フォルダ(今回だと「CompBio2025/01」)へ 忘れずにノートブックを移動させよう

### ドラッグ&ドロップ

1. 移動させたいファイルをドラッグ&ド ロップして,移動先まで持っていく.

۵	ドライブ	Q、ドライブで検索		•
+	新規	マイドライブ > Colab Notebooks 👻		e
0	候補	名前 ↑	オーナー	最終更新
4	マイドライブ	Compbio2019	自分	2019/07/16 自分
	共有ドライブ	frame_horse	自分	2018/12/19 自分
00	共有アイテム	malaria 🖿	自分	2018/12/19 自分
0	最近使用したアイテム	project_029	自分	2019/08/29 自分
☆	スター付き	subproject_013	自分	2019/04/24 自分
	ゴミ箱	subproject_016	自分	2019/10/04 自分
	保存容量	subproject_022	自分	2020/02/05 自分
	24 GB 使用	subproject_023	自分	2019/04/02 自分
		video	自分	2018/12/19 自分
		🚥 01_n 🦲 01.ipynb	自分	2020/05/02 自分
		see Of Joynh		
		02_python.ipynb	自分	2018/12/19 自分

## Googleドライブから新規作成 1. ノートブックを保存したいフォルダへ移動 例. マイドライブ $\rightarrow$ CompBio2025 $\rightarrow$ 01

「新規作成」(もしくは右クリック)から「その他」→「Google Colaboratory」 2.



# ノートブックを新規作成する(2)

### 好きな場所にノートブックを作成できる

(既存の) ノートブックを開く(1)

## Googleドライブから開く(Colab内)

1. Colabの「ファイル」

### 2. ノートブックを開く



3. Google ドライブのタブを選択

例	最近	Google ドライブ	GitHub	ፖ	ップロー	-14
ートブックを	絞り込む	Ŧ				
タイトル	6	所有者	最終更新	最終閲覧		
01.ipynb		Koji Noshita	13 分前	13 分前		Ø
Untitled2	2.ipynb	Koji Noshita	44 分前	44 分前	۵	
🝐 test-01.ip	pynb	Koji Noshita	8時間前	8 時間前	۵	
🝐 keras-un	et.ipynb	Koji Noshita	2020年2月23日	2020年2月23日		Z
🛆 mask_rc	nn.ipynb	Koji Noshita	2020年2月23日	2020年2月23日		Z
						r 5

4. 開きたいファイルを選択

## Googleドライブから開く(直接)

1. Googleドライブで開きたいファイルの あるフォルダへ移動 例) CompBio2024/01

### 2. 以下のいずれか

• 右クリック→アプリで開く→Google Colaboratory



## (既存の) ノートブックを開く(2)

## ・開きたいファイルをダブルクリック

→ Google Colaboratoryで開く

CO Google Colab	ooratory で開く│ <del>▼</del>	-
Colab Notebooks 👻		
		最終更新
		2019/04/24 自分
		2019/10/04 自分
プレビュ-	ーできません	05 自分
▶ ダウンロード	🚡 その他のアプリ	02. 自分 りを接続 19. 自分
このアイテムを表示または編集す	白分 るアプリを以下から選	 【択してください
接続済 CO Google C	自分 Colaboratory 自分	2018/12/19 自分 2018/12/19 自分

018/12/19	自分	
.018/12/19	自分	

# Colabを使うときの注意点

- ノートブックを何も操作しない(アイドル)状態だとセッションが切れる
  - セッションが切れただけであれば再接続できる。
- ・ノートブックのセッションが切れて90分経過するとインスタンスが落ちる
- インスタンスは起動後12時間立つと(セッションの有無に関係なく)落ちる
- ・同時起動できるインスタンスは3つ(CPUのみ、GPU、TPU)
  - ・ 複数のノートブックをただ単に開いても同じインタンス上で動いている
  - 共有したノートブックの閲覧だけならインスタンスへの接続は必要ない

### より詳しい情報

- こんにちはColaboratory https://colab.research.google.com/notebooks/welcome.ipynb
- Colaboratory よくある質問 https://research.google.com/colaboratory/faq.html
- 【秒速で無料GPUを使う】深層学習実践Tips on Colaboratory | Qiita https://giita.com/tomo\_makes/items/b3c60b10f7b25a0a5935
- 【はやわかり+保存版】Google Colaboratoryチートシート https://giita.com/tomo\_makes/items/ec12dd61e4ced0a3dd66

あまり気にしなくても良い補足

- セッション:ここではローカルマシン(皆さんが 使っているPC)とインスタンスとの接続のこと.
- インスタンス:ここではGoogle Cloud上の仮想 マシンのこと. 通常はColabはこのインスタンス 上で動作している. つまり, Pythonのコードは (皆さんのPC上では実行されておらず) クラウド にあるインスタンスで実行され、その結果をブラ ウザを介して見ている.



- ・テキストセル(text cell)に文章や数式を入力することができる
  - ・
     文章はMarkdown記法が使える
  - 数式はLaTeXが使える



テキストの記述

# 見出し(#):Markdownでは# をつけることで見出しにでき る.##や###など#を重ねるこ

課題への取り組み・提出

れば、より知りたいと思った点を教えて、

2.今後扱って欲しい内容を挙げる(複数可)

3. 「Hello, World!」 プログラムを作成せよ.

ハマってます、など)

5.その他質問、感想、要望をどうぞ、

本日の課題 ノーマル

- 1.今回の授業でよくわからなかった点を述べる. 特になけ
- 4.Slackの「#自己紹介」チャンネルに自分の自己紹介を投 稿する、(例、~~です、趣味は~~です、最近~~に

- !注意:第1回課題提出の締め切りは、第2回の前日23:59まで.
  - 課題をGitHub Classroomにて提出すること

# GitHub Classroomによる課題の配布

## 課題配布URLへアクセス



### GitHub Classroom by GitHub would like permission to

Cancel	Authorize GitHub
earn more about GitHub Classroom.	
Plan (read) View your subscription plan of	n GitHub
Email addresses (read) View your email addresses	
Resources on your account	
<ul> <li>Act on your behalf</li> <li>(?) Learn more</li> </ul>	
Know which resources you	ı can access
Verify your GitHub identity	/ (noshita-c01)

初回のみアクセス許可 の承認が必要です

ku-compbio-2025

### Accept the assignment assignment-01

Once you accept this assignment, you will be granted access to the assignment-01-noshita-c01 repository in the ku-compbio-2025 organization on GitHub.



課題を受け取ると, ku-compbio-2025/assignment-NN-XXX に課題レポジトリが作成される NNは課題番号,XXXはアカウント名.



## 2. 課題レポジトリ

Accept this assignment



## 課題レポジトリへアクセス

🔳 🎧 ku-	compbio-2025 / assignment-01-noshita-c01 ¥		Q Type [] to search	8 •   + • O h @ (
↔ Code ⊙	Issues 📫 Pull requests 📀 Actions 🖽 Projects 🕕	Security 🗠 Insights		
	sorked from ku-compbio-2025/ku-compbio-2025-assignment-01-assignme	<u>nt-01</u>	⊙ Watch 0	+ to Fork 0 + ☆ Star 0 +
	ᢞ main → 양 1 Branch 🛇 0 Tags	Q. Go to file	t Add file + Code +	About
	This branch is 1 commit ahead of main .		11 Contribute * 😳 Sync fork *	ku-compbio-2025-assignment-01- assignment-01 created by GitHub Classroom
	github-classroom[bot] add deadline		577c874 · 52 minutes ago 🕓 2 Commits	🛱 Readme
	🗋 01.ipynb	mmit	5 days ago	Activity
	README.md     add dea	dline	52 minutes ago	☆ 0 stars
			Ø 🗉	<ul> <li>O watching</li> <li>♀ 0 forks</li> </ul>
	📋 Review the assignment due date			Releases No releases published <u>Create a new release</u>
	第1回課題			Packages
	以下の課題に取り組み, @1.ipynb 完成させてください			No packages published Publish your first package
	課題			Languages
	1. 今回の授業でよくわからなかった点を述べる。特に	なければ、より知りたい	と思った点を教えて	Junyter Notebook 100 0%



## (既存の) ノートブックを開く(3) GitHubから開く 1. ファイルを開く→GitHub 2. 検索→リポジトリを選択

ノートブック							
例	>	<ul> <li>GitHub URL を入力するか、組織またはユー</li> <li>ku-compbio-2025</li> </ul>	ザーで検导	索します ――――	٩	✓ 非公開リ	ポジトリを含める
最近	>	リポジトリ: 🔼		ブランチ:			
Google ドライ ブ	>	ku-compbio-2025/assignment- 01-noshita	•	main		•	
GitHub	>						
アップロード	>	パス					
		O1.ipynb					• 🖸

Aut	horize Col	aborato	D V
Colaboratory b wants to access	y Google Colab your noshita-c01	ooratory account	
Repositories	eccess		~
Cancel		Authorize	googlecolab
http	Authorizing will r os://colab.researc	edirect to h.google.com	
Not owned or operated by GitHub	Created 8 years ago	, D	More than 1K GitHub users
	Learn more abo	ut OAuth	

「非公開リポジトリを含める」 にチェック

初回はアクセス許可に承諾

ノートブックを開	<		
例	GitHub URL を入力するか、組織またはユーザーで検索し ku-compbio-2025	ر ます <b>Q</b>	▶ 非公開リポジトリを含める
最近 >	リポジトリ: 🔼	ブランチ: 🚺	
Google ドライ ブ	ku-compbio-2025/assignment- 01-noshita	main	•
GitHub >			
アップロード 🔉	パス		
	O1.ipynb		

- ・「ku-compbio-2025」で検索
- ・先程受け取った課題リポジトリを選択
  - ・ku-compbio-2025/assignment-NN-XXX NNは課題番号,XXXはアカウント名
- ・01.ipynbを開く



課題提出用ノートブックの作成(1)

## 課題を作成して、GitHubにて提出する

## Googleドライブに一旦保存して もOK

→ 何度も保存する場合はこちら の方が便利

ファイル → ドライブにコピーを保存

GitHub で表示	
ドライブの新しいノートブック	
ノートブックを開く	₩/C
ノートブックをアップロード	
名前の変更	
ドライブにコピーを保存	
コピーを GitHub Gist として保存	
保存	H/C
変更履歴	
ダウンロード	
印刷	ዤ/C







課題提出用ノートブックの作成(2)

### 課題を作成して、GitHubにて提出する

<b>티</b> コメン ヘルプ <u>すべての変更を保存しました</u>	٢	*	共有	¢	Ę	
	接続	•	1	編集	^	
作成せよ.						
この説明があるとなお良い(その場合	は	サ	ブ			
サブセクション(###)などにする)						
・ルに自分の自己紹介を投稿する. (例. ~~です ど)	└.	取吁	<b>⊧は</b> ~	~~7	<del>ر</del> ،	

ノートブックな

## GitHubへ保存

1. ファイル→保存

	GitHub で表示	
CO ファイル編集 GitHubから	ドライブの新しいノートブック ノートブックを開く ノートブックをアップロード	೫/Ctrl+0
開いたファ	名前の変更	
イルのままの担合	ドライブにコピーを保存 コピーを GitHub Gist として保存	
	保存 変更履歴	೫/Ctrl+
	ダウンロード 印刷	ا \$#/Ctrl+I

🛆 01.ipynb CO ファイル 編集 Googleドラ イブにコピー を保存した場 合

GitHubに コピーを保存 する

ドライブで探す	
Playground モードで開く	
ドライブの新しいノートブック	
ノートブックを開く	쁐/Ctrl+O
ノートブックをアップロード	
名前の変更	
移動	
ゴミ箱に移動	
ドライブにコピーを保存	
コピーを GitHub Gist として保存	
GitHub にコピーを保存	
保存	육/Ctrl+S
版を保存して固定	೫/Ctrl+M S
変更履歴	
ダウンロード	Þ.
印刷	윎/Ctrl+P

∠. ノ ゾ ゴ ノレ→☆	下1子
GitHub に保存	
リポジトリ: 🔼	ブランチ: 🔼
ku-compbio-2025/assignment- 01-noshita	main 🝷
- ファイルパス* 01.ipynb	
ー コミット メッセージ ―― Hello, Worldプログラムの作成	
□ Colab へのリンクを含める	
	キャンセル

- ・先程受け取った課題リポジトリを選択
- コミットメッセージを入力

を保存する(2)	)
----------	---

コミットメッセージについて Gitではファイルの変更履歴をコミットとい う単位で管理する。コミットメッセージは各 コミットに対する説明文にあたる. 本演習では、どのような変更を加えたかを 簡潔に入力することを推奨する。例えば、 「変更した」だけよりは「指数増殖をシミュ レーションする関数を追加した」のようなあ る程度具体的な説明が望ましい。どのように 記述すると良いかのベストプラクティスなども ある. 例えば、Conventional Commits https://www.conventionalcommits.org/

ku-compbio-2025/assignment-NN-XXX

NNは課題番号、XXXはアカウント名.

・ファイルパスに開いたファイル(01.ipynb)を指定



# 課題の提出

ి main 👻 ్రి 1 Branch 📀 0 Tags	Q Go to file	t Add file 👻	<> Code
This branch is 3 commits ahead of ku-compbio-20	025/ku-compbio-2025-assignment-01-assi	gnment-01:main .	
이 Contribute - C Sync fork -			
5 noshita Hello, Worldプログラムの作成		f062dee · 2 hours ago	🕓 4 Commit
ិ 01.ipynb	Hello, Worldプログラムの作成		2 hours ag
B README.md	add deadline		5 days ag
D README			0 ::
📋 Review the assignment due date			
第1回課題			

## 課題レポジトリが更新されている

# 課題用レポジトリヘコミット(ここではファイルを保存)すること

## で、課題は提出状態になる、期限内であれば、何度でも変更可能

🚯 noshita Hello, Worldプログラムの作成		f062dee · 2 hours ago
Preview Code	Blame 128 lines (128 loc) · 2.69 KB	Raw C 2
	第1回課題	
	1.今回の授業でよくわからなかった点を述べる.特になけれは と思った点を教えて.	、より知りたい
	GitHub Classroomの使い方がよくわからない	
	2.今後扱って欲しい内容を挙げる(複数可).	
	3. 「Hello, World!」プログラムを作成せよ.	
In [1]	<pre>print("Hello, World!")</pre>	
	Hello, World!	
	<b>4. Slack</b> の「#自己紹介」チャンネルに自分の自己紹介を投稿し す. 趣味は~~です. 最近~~にハマってます. など), その	,(例. ~~で ) <b>URL</b> を記載して.

## 課題レポジトリから確認できる



### Googleなどで調べる

- 状況にあったキーワードを2~5個程度を考え、検索する
  - •例) Python Colab ノートブック ダウンロード.
- エラーメッセージを読む、検索する。
  - •例)TypeError~~.

Slackの「#質問」チャンネルを使う

- ・

   ・
   質問する
  - ど、~にならない.
- 他の人の質問に答える

(そもそも何で詰まってるのか説明できないと,答えようがないので)まずは自分で調べる・考える. それでも無理そうだったら野下やTAにどんどん気軽に質問する(特に最初はしょうもないミスに時間を取 られがち、無駄に数時間悩むより質問する。)、自分ができていたら困っている人を助けよう。

質問・トラブルシューティング

例)インストールが~で止まる。~の図がきれいに表示されない。~をやってみたけ

例) ~にチェックを入れるといいよ、途中の~の計算ができてないんじゃない?

サンプルコード,スライド,資料などに間違いや不正確な内容がある可能性がある. これらを本演習では総じて「バグ」と呼ぶ、もし、バグを見つけたら報告しよう、

## Slackの「#バグ報告」チャンネルを使う

- ・バグの指摘
- バグの指摘と原因の特定
- ・既に指摘されているバグの修正案を提案
  - 例) ~のバグだけど、~を~に修正すると正しい結果が得られるよ.
- ・バグの指摘と修正案の提案
  - に修正すると正しい結果が得られる。

バグ報告をしてもらえるとコードや資料の改善に繋がり非常に助かる。 バグ報告にボーナスポイントあり

バグ報告

• 例)サンプルコードの~を実行するけど、(予想される結果)~にならない.

例) ~を実行するけど、(予想される結果) ~にならない、~のせいだ。

例) ~を実行するけど、(予想される結果) ~にならない、~のせいだ、~を~

れば、より知りたいと思った点を教えて、

2.今後扱って欲しい内容を挙げる(複数可)

3. 「Hello, World!」 プログラムを作成せよ.

ハマってます、など)

5.その他質問、感想、要望をどうぞ、

本日の課題 ノーマル

- 1.今回の授業でよくわからなかった点を述べる. 特になけ
- 4.Slackの「#自己紹介」チャンネルに自分の自己紹介を投 稿する、(例、~~です、趣味は~~です、最近~~に

- !注意:第1回課題提出の締め切りは、第2回の前日23:59まで.
  - 課題をGitHub Classroomにて提出すること

# 第2回: Pythonの基本的な使い方と 数理生物学演習で使う数学の復習 4月21日

- ・ベクトル、行列の演算



# 復習推奨

固有値・固有ベクトルの計算 微分方程式の変数分離による解法