

DeepLabCut :: CHEAT SHEET



Introduction

DeepLabCut is an open source automated animal pose estimation software using deep learning.

Many excellent resources exist for learning how to install and use DeepLabCut – this is only intended as a quick overview. For more, see <https://mouselab.org/deeplabcut> and <https://github.com/DeepLabCut/DeepLabCut>

1. Use a **layout** that flows and makes it easy to zero in on specific topics.
2. Use **visualizations** to explain concepts quickly and concisely.
3. Use visual elements to make the sheet **scannable**.

Command overview

Operations	Command
Open IPython and import DeepLabCut (Step 1)	<code>ipython</code> <code>import deeplabcut</code>
Create a new project (Step 2)	<code>deeplabcut.create_new_project('project_name', 'experimenter', ['path of video 1', 'path of video2', ...])</code>
Set a config_path variable for ease of use (Step 3)	<code>config_path = '/yourdirectory/project_name/config.yaml'</code>
Extract frames (Step 4)	<code>deeplabcut.extract_frames(config_path, mode='automatic', algo='kmeans', crop=True/False)</code>
Label frames (Steps 5 and 6)	<code>deeplabcut.label_frames(config_path)</code>
Check labels (optional)(Step 7)	<code>deeplabcut.check_labels(config_path)</code>
Create training dataset (Step 8)	<code>deeplabcut.create_training_dataset(config_path)</code>
Train the network (Step 9)	<code>deeplabcut.train_network(config_path)</code>
Evaluate the trained network (Step 10)	<code>deeplabcut.evaluate_network(config_path)</code>
Video analysis and plotting results (Step 11)	<code>deeplabcut.analyze_videos(config_path, ['path of video 1 or folder', 'path of video2', ...])</code>
Video analysis and plotting results (Step 12)	<code>deeplabcut.plot_trajectories(config_path, ['path of video 1', 'path of video2', ...])</code>
Video analysis and plotting results (Step 13)	<code>deeplabcut.create_labeled_video(config_path, ['path of video 1', 'path of video2', ...])</code>
Refinement: extract outlier frames (Step 14)	<code>deeplabcut.extract_outlier_frames(config_path, ['path of video 1', 'path of video 2'])</code>
Refine labels (Step 15)	<code>deeplabcut.refine_labels(config_path)</code>
Combine datasets (Step 16)	<code>deeplabcut.merge_datasets(config_path)</code>

Introduction

DeepLabCut is an open source automated animal pose estimation software using deep learning.

Many excellent resources exist for learning how to install and use DeepLabCut – this is only intended as a quick overview. For more, see <https://mouselab.org/deeplabcut> and <https://github.com/DeepLabCut/DeepLabCut>

1. Use a **layout** that flows and makes it easy to zero in on specific topics.
2. Use **visualizations** to explain concepts quickly and concisely.
3. Use visual elements to make the sheet **scannable**.

Command overview

Operations	Command
Open IPython and import DeepLabCut (Step 1)	<code>ipython import deeplabcut</code>
Create a new project (Step 2)	<code>deeplabcut.create_new_project('project_name','experimenter', ['path of video 1','path of video2',...])</code>
Set a config_path variable for ease of use (Step 3)	<code>config_path = '/yourdirectory/project_name/config.yaml'</code>
Extract frames (Step 4)	<code>deeplabcut.extract_frames(config_path, mode='automatic', algo='kmeans', crop=True/False)</code>
Label frames (Steps 5 and 6)	<code>deeplabcut.label_frames(config_path)</code>
Check labels (optional)(Step 7)	<code>deeplabcut.check_labels(config_path, visualizeindividuals=True/False)</code>
Build skeleton (Step 8)	<code>deeplabcut.SkeletonBuilder(config_path)</code>
	<code>deeplabcut.cropimagesandlabels(config_path)</code>
Create training dataset (Step 8)	<code>deeplabcut.create_multianimaltraining_dataset(config_path, allow_growth=True)</code>
Train the network (Step 9)	<code>deeplabcut.train_network(config_path)</code>
Evaluate the trained network (Step 10)	<code>deeplabcut.evaluate_network(config_path)</code>
Video analysis and plotting results (Step 11)	<code>deeplabcut.analyze_videos(config_path,['path of video 1 or folder','path of video2',...])</code>
Video analysis and plotting results (Step 12)	<code>deeplabcut.plot_trajectories(config_path,['path of video 1', 'path of video2',...])</code>
Video analysis and plotting results (Step 13)	<code>deeplabcut.create_labeled_video(config_path,['path of video 1', 'path of video2',...])</code>
Refinement: extract outlier frames (Step 14)	<code>deeplabcut.extract_outlier_frames(config_path,['path of video 1', 'path of video 2'])</code>
Refine labels (Step 15)	<code>deeplabcut.refine_labels(config_path)</code>
Combine datasets (Step 16)	<code>deeplabcut.merge_datasets(config_path)</code>

Three Column Layout: : CHEAT SHEET



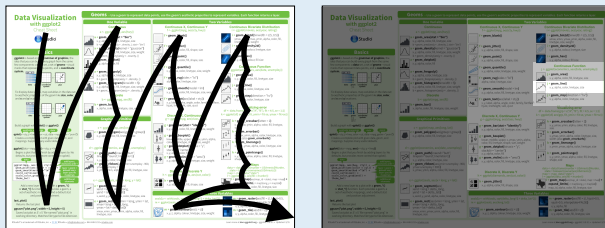
Basics

Thank you for making a new cheatsheet for R! These cheatsheets have an important job:

Cheatsheets make it easy for R users to look up useful information.

Remember that the best cheatsheets are **visual**—not written—documents. Whenever possible use visual elements to make it easier for readers to find the information they need.

1. Use a **layout** that flows and makes it easy to zero in on specific topics.



2. Use **visualizations** to explain concepts quickly and concisely.

summary function

3. Use visual elements to make the sheet **scannable**.

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

4. Use visual **emphasis** (like color, size, and font weight) to make important information easy to find.

dplyr::lag() - Offset elements by 1
dplyr::lead() - Offset elements by -1

COPYRIGHT

Each cheatsheet should be licensed under the creative commons license.

To license the sheet as creative commons, put CC'd by <your name> in the small print at the bottom of each page and link it to <http://creativecommons.org/licenses/by/4.0/>

YOUR LOGO
(optional)

Layout Suggestions

Use headers, colors, and/or backgrounds to **separate or group together sections**.

Section 1

Section 2

Section 3

Create a visual hierarchy. Help users navigate the page with titles, subtitles, and subsubtitles

Title

SUBTITLE

SUBSUBTITLE

Manipulate Variables

Quickly identify content with a **package hexsticker** (if available)

Fit sections to content. Try several different layouts.

Use numbers or arrows to link sections if the order/**flow** is confusing.

Logistics

FONTS

This template uses several fonts: **Helvetica Neue**, **Menlo**, **Source Sans pro**, which you can acquire for free here, www.fontsquirrel.com/fonts/source-sans-pro, and **Font Awesome**, which you can acquire here, fontawesome.github.io/Font-Awesome/get-started/

To use a **font awesome** icon, copy and paste one from here fontawesome.github.io/Font-Awesome/cheatsheet/. Then set the text font to font awesome.

KEYNOTE

I make my cheatsheets in **Apple Keynote**, and not latex or R Markdown, because presentation software makes it much easier to tweak the visual appearance of a document

KEYNOTE TIPS

- **Select multiple elements** by holding down shift and then selecting each. Click on a selected element before letting go of shift to unselect it.
- To **group elements together**. Select them all, then click Arrange > Group
- To **evenly space multiple objects**, select them all then Right Click > Align objects or Right Click > Distribute objects
- Click on a table, then visit Format > Table > Row and Column Size to make **even width rows/columns**.

Useful Elements

CODE

Where possible, use **code that works** when run.

```
ggplot(mpg, aes(hwy, cty)) +  
  geom_point(aes(color = cyl)) +  
  geom_smooth(method = "lm")
```

word balloons

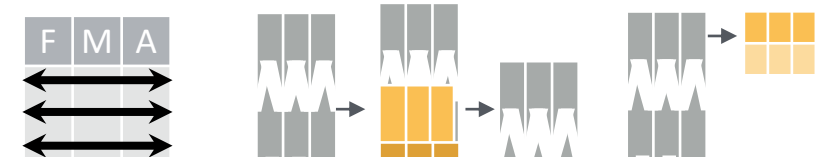
can help explain code

ICONS

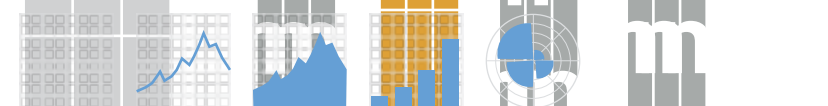


These are just font awesome characters

MOCK TABLES



MOCK GRAPHS



TABLES

