

# Part 1.2: Installing Software and Package Managers



Dana L Carper and Travis J Lawrence  
Quantitative and Systems Biology  
University of California, Merced

# Computer Programs

- Written in programming languages
- Are not always easily installed
- Scientific programs usually come as source code
  - Contains general information to run on any computer
  - Needs to be compiled and installed

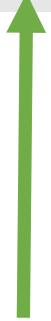
# Why compile software?

- Customizes for you system
  - optimized for your computer/cluster
- Can change where things are installed
- Can have multiple versions of a program:
  - In case you are using older software

# Important files before you get started

```
$ ls
```

Prompt



# Important files before you get started

```
$ ls
```

Prompt

list what is in  
that folder

# Important files before you get started

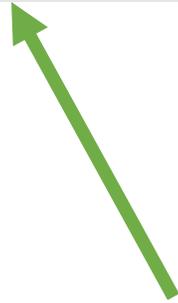
```
$ ls  
README  
INSTALL
```



Files that  
contain  
information on  
installing the  
software

# Important files before you get started

```
$ ls  
README  
INSTALL  
$ less README
```



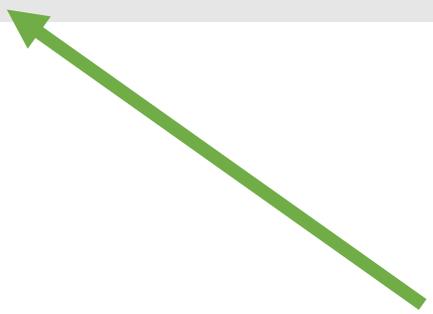
open this file  
and look at it in  
the terminal

# Important files before you get started

```
$ ls  
README  
INSTALL  
$ less README
```



open this file  
and look at it in  
the terminal



File you want to  
look at

# Important files before you get started

```
hisat2 — less README.md — 80x24
# hisat2
HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (whole-genome, transcriptome, and exome sequencing data) to a population of human genomes (as well as to a single reference genome). Based on an extension of BWT for a graph [1], we designed and implemented a graph FM index (GFM), an original approach and its first implementation to the best of our knowledge. In addition to using one global GFM index that represents general population, HISAT2 uses a large set of small GFM indexes that collectively cover the whole genome (each index representing a genomic region of 56 Kbp, with 55,000 indexes needed to cover human population). These small indexes (called local indexes) combined with several alignment strategies enable effective alignment of sequencing reads. This new indexing scheme is called Hierarchical Graph FM index (HGFM). We have developed HISAT2 based on the HISAT [2] and Bowtie 2 [3] implementations. See the [HISAT2 website](http://ccb.jhu.edu/software/hisat2/index.shtml) for more information.

A few notes:

1) HISAT2's index (HGFM) size for the human reference genome and 12.3 million common SNPs is 6.2GB. The SNPs consist of 11 million single nucleotide polymorphisms, 728,000 deletions, and 555,000 insertions. Insertions and deletions used in this index are small (usually <20bp). We plan to incorporate structural variations:
```

# Compiling and Installing from Source Code

- Once you look at README/INSTALL files
- You install software using three commands (usually):
  - `configure`
  - `make`
  - `make install`

# Configure

- Configure sets up the program for your specific system

```
$ ./configure
```

# Configure

- Configure sets up the program for your specific system

```
$ ./configure
```

Prompt

Means do the command to a file in the folder I am currently in

# Configure

- Configure sets up the program for your specific system

```
$ ./configure
```

Prompt

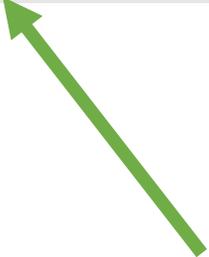
Means do the command to a file in the folder I am currently in

The actual command to configure, will look for a configure script

# Configure

- Configure sets up the program for your specific system

```
$ ./configure  
$ ls
```

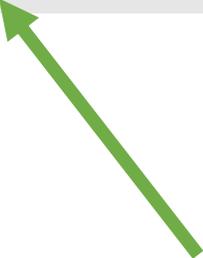


list what is in  
that folder

# Configure

- Configure sets up the program for your specific system

```
$ ./configure  
$ ls  
Makefile
```

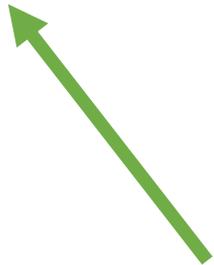


what is created  
from the  
configure  
command

# Configure

- Configure sets up the program for your specific system

```
$ ./configure  
$ ls  
Makefile  
Makefile.in
```



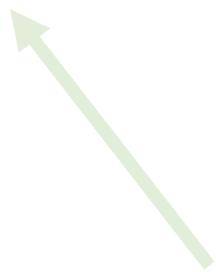
Some programs  
come with a  
template file

# Configure

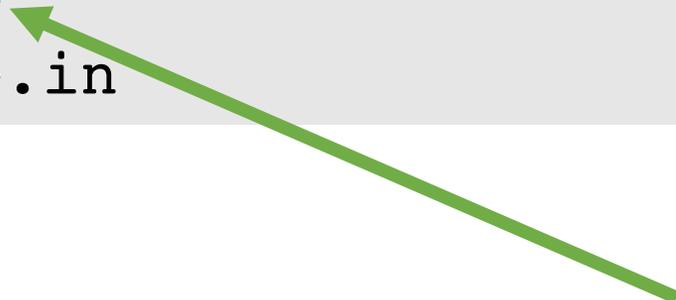
- Configure sets up the program for your specific system

```
$ ./configure  
$ ls  
Makefile  
Makefile.in
```

Some programs  
come with a  
template file



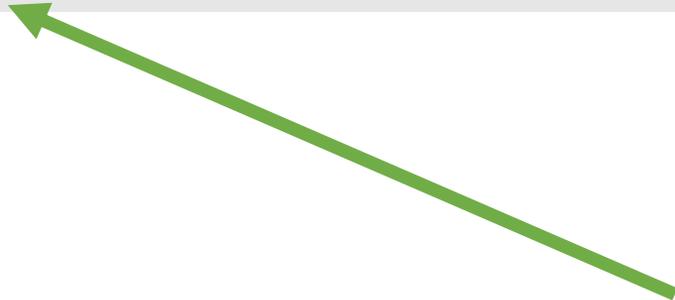
This file is  
customized to  
your system



# Make

- Make will finish building the program from source code into a usable form

```
$ ./configure  
$ ls  
Makefile
```

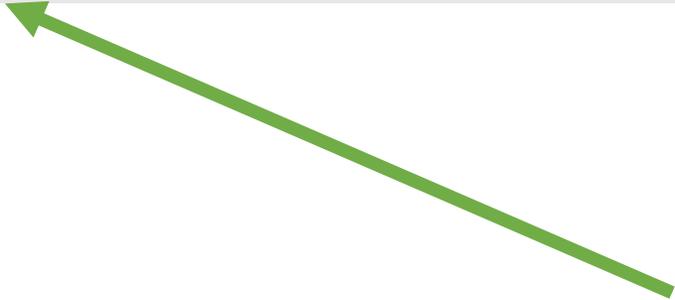


Uses what is in  
this file

# Make

- Make will finish building the program from source code into a usable form

```
$ ./configure  
$ ls  
Makefile  
$ make
```

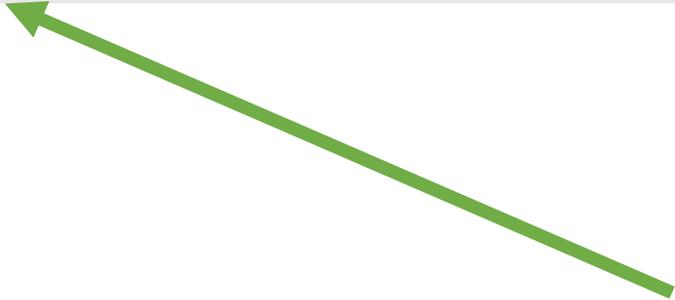


Builds the  
program

# Make

- Make will finish building the program from source code into a usable form

```
$ ./configure  
$ ls  
Makefile  
$ make
```

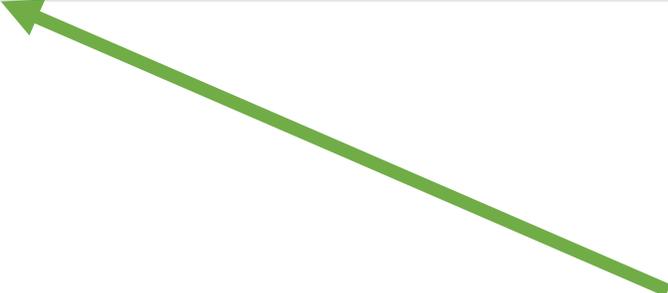


After this you get a  
useable program in the  
folder you are currently in

# Make install

- Will copy the built program, and its libraries and documentation, to the correct locations (in your \$PATH)

```
$ ./configure  
$ ls  
Makefile  
$ make  
$ make install
```



After should be able to use  
the program regardless of  
where you are in your  
folders

# Not all programs need all three commands

- Commonly you need all three commands:
  - configure
  - make
  - make install
- Some programs come pre-configured:
  - only need to run make and make install
- Some programs you have to do the final install:
  - no make install
  - You have to move programs to your \$PATH

# What is a package manager?

- A collection of software tools that automates:
  - installing computer programs
  - upgrading computer programs
  - configuring computer programs
  - removing computer programs
- Examples:
  - Homebrew
  - Perlbrew
  - Macports

# Using Package Managers

- Example using Homebrew

```
$ brew update
```

Prompt

Telling the computer  
you want brew (the  
package manager) to  
do something

# Using Package Managers

- Example using Homebrew

```
$ brew update
```

Prompt

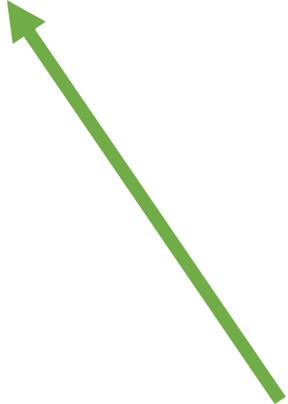
Telling the computer you want brew (the package manager) to do something

Update brew itself to the newest version

# Using Package Managers

- Example using Homebrew

```
$ brew update  
$ brew upgrade
```

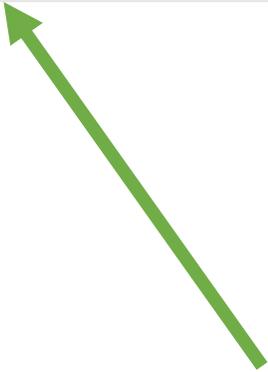


Upgrade any  
programs to the  
newest versions

# Using Package Managers

- Example using Homebrew

```
$ brew update  
$ brew upgrade  
$ brew cleanup
```

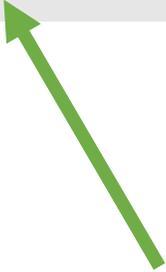


Uninstall old  
versions of  
programs

# Using Package Managers

- Example using Homebrew

```
$ brew update  
$ brew upgrade  
$ brew cleanup  
$ brew install git
```



Telling brew you want  
to install a program  
and any  
dependencies

# Using Package Managers

- Example using Homebrew

```
$ brew update  
$ brew upgrade  
$ brew cleanup  
$ brew install git
```



Telling brew you want  
to install a program  
and any  
dependencies



Dependencies are any  
other scripts or  
programs that are  
required to run

# Using Package Managers

- Example using Homebrew

```
$ brew update  
$ brew upgrade  
$ brew cleanup  
$ brew install git
```



Telling brew you want to install a program and any dependencies



The program you want to install

# Anaconda

- Is a virtual environment
- Growing repository of bioinformatics software
- Manages different package versions

# Anaconda

- Example using pip

```
$ pip install biopython
```



package  
manager  
specifically for  
python  
packages

# Anaconda

- Example using pip

```
$ pip install biopython
```

package  
manager  
specifically for  
python  
packages



Action  
you want  
pip to do



# Anaconda

- Example using pip

```
$ pip install biopython
```

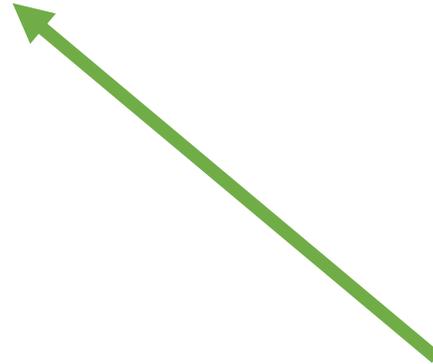
package  
manager  
specifically for  
python  
packages



Action  
you want  
pip to do



Python package  
you want  
installed



# Where are programs stored?

- Programs are usually stored within repositories:
  - cpan: programs written in the programming language perl
  - PyPI : programs written in the programming language python
  - github: generic repository for any program
  - CRAN: programs written in R scripting language

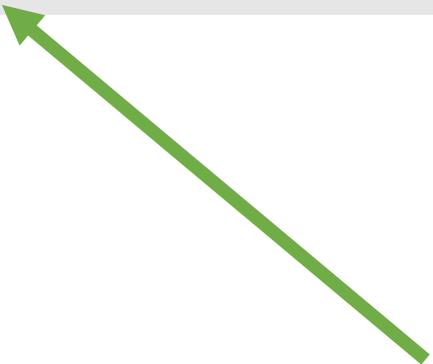
# Github

- A web-based repository
- Used for development of programs with many collaborators
- Often has the newest versions of software

# Getting files from git hub

- Requires the program git

```
$ brew install git
```



git can be installed  
through a package  
manager

# Getting files from git hub

The screenshot displays the GitHub interface for the repository `infphilo/hisat2`. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Gist. Below this, the repository name `infphilo / hisat2` is shown, along with statistics: 32 Watchers, 110 Stars, and 29 Forks. Navigation tabs include Code (selected), Issues (41), Pull requests (0), Projects (0), Wiki, and Insights.

The main content area is titled "Graph-based alignment (Hierarchical Graph FM index)". Below the title, repository statistics are shown: 775 commits, 10 branches, 1 release, 9 contributors, and GPL-3.0 license. A progress bar is visible below these statistics.

Below the progress bar, there are buttons for "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download".

The repository content is listed below, showing folders and their commit history:

Folder	Commit Message	Time Ago
<code>doc</code>	HISAT2 2.0.5 release on Nov 4, 2016	7 months ago
<code>evaluation</code>	Support for Visual Studio on Windows system was implemented by Nigel ...	7 months ago
<code>example</code>	Added a new example	2 years ago
<code>genotype_scripts</code>	Changed extract_var script to work on CYP genes - not finished yet	11 months ago
<code>hisat2.xcodeproj</code>	Set the default value of --max-seeds is at least 5	7 months ago
<code>hisat2_modules</code>	in the middle of HLA allele assembly	7 months ago
<code>hisat2_test</code>	Updated test scripts	2 years ago
<code>li_hla</code>	.	a year ago

# Getting files from git hub

The screenshot shows the GitHub repository page for `infphilo/hisat2`. The repository has 775 commits, 10 branches, 1 release, 9 contributors, and is licensed under GPL-3.0. The commit history table is as follows:

Commit	Message	Time
infphilo	Updated the Manual for --max-seeds	Latest commit 78608ae on Nov 5, 2016
doc	HISAT2 2.0.5 release on Nov 4, 2016	7 months ago
evaluation	Support for Visual Studio on Windows system was implemented by Nigel ...	7 months ago
example	Added a new example	2 years ago
genotype_scripts	Changed extract_var script to work on CYP genes - not finished yet	11 months ago
hisat2.xcodeproj	Set the default value of --max-seeds is at least 5	7 months ago
hisat2_modules	in the middle of HLA allele assembly	7 months ago
hisat2_test	Updated test scripts	2 years ago
li_hla	.	a year ago

Where you can  
get the web  
address

# Getting files from git hub

The screenshot shows the GitHub interface for the repository 'infphilo/hisat2'. At the top, there are navigation links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. Below this, the repository name 'infphilo / hisat2' is displayed along with statistics: 32 Watchers, 110 Stars, and 29 Forks. A navigation bar includes 'Code', 'Issues (41)', 'Pull requests (0)', 'Projects (0)', 'Wiki', and 'Insights'. The main content area is titled 'Graph-based alignment (Hierarchical Graph FM index)'. A summary bar shows 775 commits, 10 branches, 1 release, 9 contributors, and GPL-3.0 license. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table is visible, with a 'Clone with HTTPS' dialog box overlaid on top. The dialog provides the URL 'https://github.com/infphilo/hisat2.git' and options to 'Open in Desktop' or 'Download ZIP'.

775 commits   10 branches   1 release   9 contributors   GPL-3.0

Branch: master   New pull request   Create new file   Upload files   Find file   Clone or download

infphilo Updated the Manual for --max-seeds

File	Commit Message	Time Ago
doc	HISAT2 2.0.5 release on Nov 4, 2016	
evaluation	Support for Visual Studio on Windows system was im	
example	Added a new example	
genotype_scripts	Changed extract_var script to work on CYP genes - not finished yet	11 months ago
hisat2.xcodeproj	Set the default value of --max-seeds is at least 5	7 months ago
hisat2_modules	in the middle of HLA allele assembly	7 months ago
hisat2_test	Updated test scripts	2 years ago
li_hla	.	a year ago
msvcc	Changed MSVCC to msvcc and removed typeof	7 months ago

Clone with HTTPS   Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/infphilo/hisat2.git

Open in Desktop   Download ZIP

# Getting files from git hub

The screenshot shows the GitHub interface for the repository 'infphilo/hisat2'. At the top, there are navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. Below the repository name, there are statistics for 'Watch' (32), 'Star' (110), and 'Fork' (29). The main content area shows a list of commits, with the most recent one by 'infphilo' titled 'Updated the Manual for --max-seeds'. A 'Clone or download' button is visible, which has opened a dialog box. The dialog box contains the text 'Clone with HTTPS' and 'Use Git or checkout with SVN using the web URL.' Below this, the URL 'https://github.com/infphilo/hisat2.git' is displayed in a text input field. A green arrow points from the text 'the web address you need to copy' to this URL field. Other options in the dialog include 'Open in Desktop' and 'Download ZIP'.

Commit	Message	Time
infphilo	Updated the Manual for --max-seeds	
	doc HISAT2 2.0.5 release on Nov 4, 2016	
	evaluation Support for Visual Studio on Windows system was im	
	example Added a new example	
	genotype_scripts Changed extract_var script to work on CYP genes - not finished yet	11 months ago
	hisat2.xcodeproj Set the default value of --max-seeds is at least 5	7 months ago
	hisat2_modules in the middle of HLA allele assembly	7 months ago
	hisat2_test Updated test scripts	2 years ago
	li_hla .	a year ago
	msvcc Changed MSVCC to msvcc and removed typeof	7 months ago

the web  
address you  
need to copy

# Getting files from git hub

- Clone repository

```
$ brew install git  
$ git clone https://github.com/infphilo/hisat2.git
```

Telling the  
computer you  
want to use the  
program git



# Getting files from git hub

- Clone repository

```
$ brew install git  
$ git clone https://github.com/infphilo/hisat2.git
```

Telling the  
computer you  
want to use the  
program git

Telling git you  
want to copy  
this

# Getting files from git hub

- Clone repository

```
$ brew install git  
$ git clone https://github.com/infphilo/hisat2.git
```

Telling the  
computer you  
want to use the  
program git

Telling git you  
want to copy  
this

The web  
address of the  
program you  
want

# Getting files from git hub

- Clone repository

```
$ brew install git  
$ git clone https://github.com/infphilo/hisat2.git
```

Telling the  
computer you  
want to use the  
program git

Telling git you  
want to copy  
this

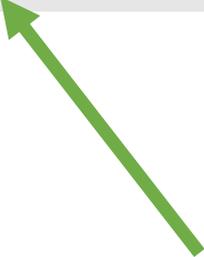
The web  
address of the  
program you  
want

copies to the  
folder you are  
currently in

# Getting files from git hub

- Clone repository

```
$ brew install git  
$ git clone https://github.com/infphilo/hisat2.git  
$ ls
```

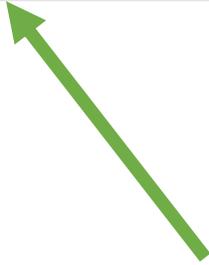


list what is in  
that folder

# Getting files from git hub

- Clone repository

```
$ brew install git
$ git clone https://github.com/infphilo/hisat2.git
$ ls
Applications      Documents Pictures
Bin                Downloads hisat2
```



What you just  
copied from github