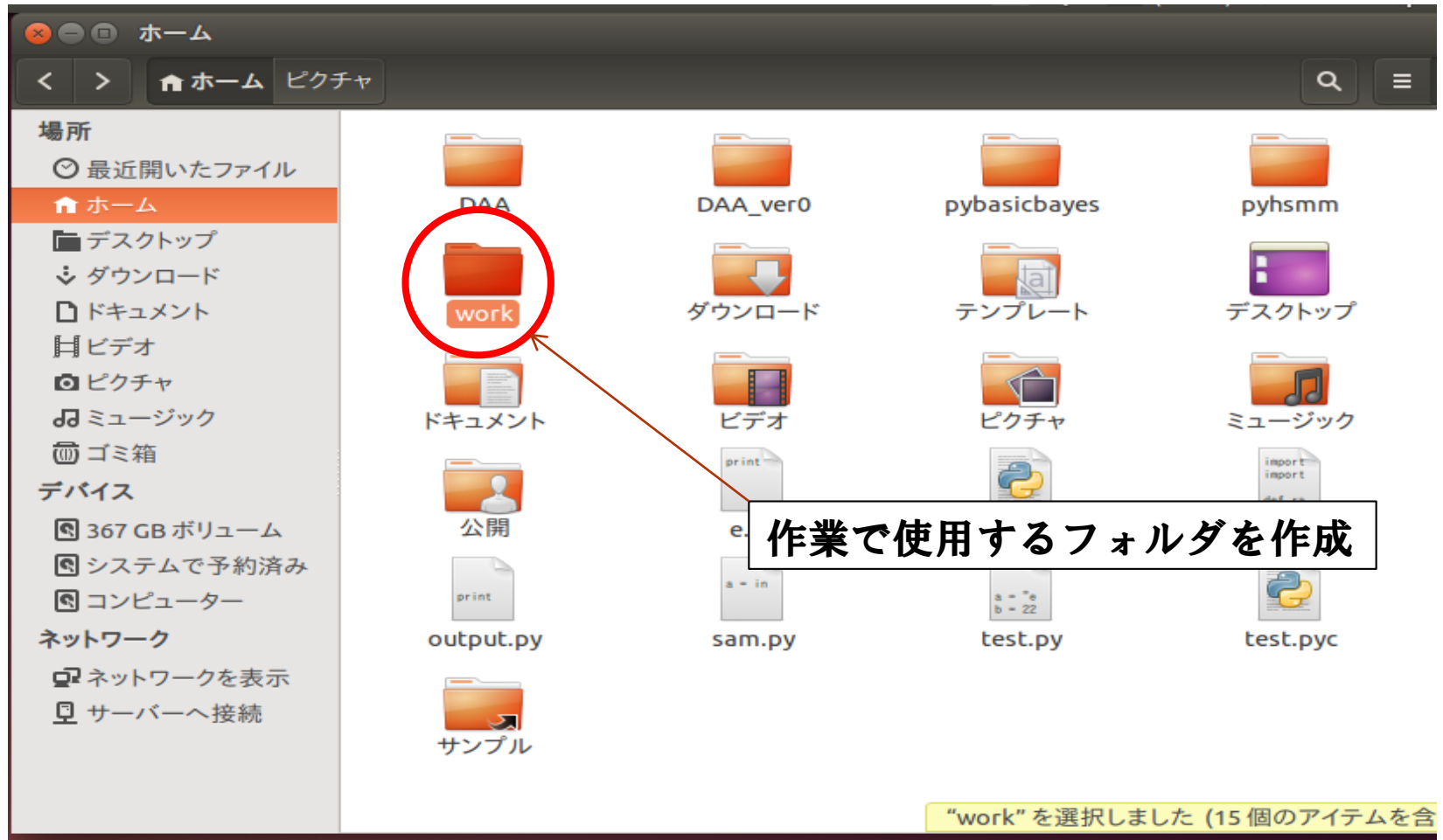


LST勉強会(第一回)

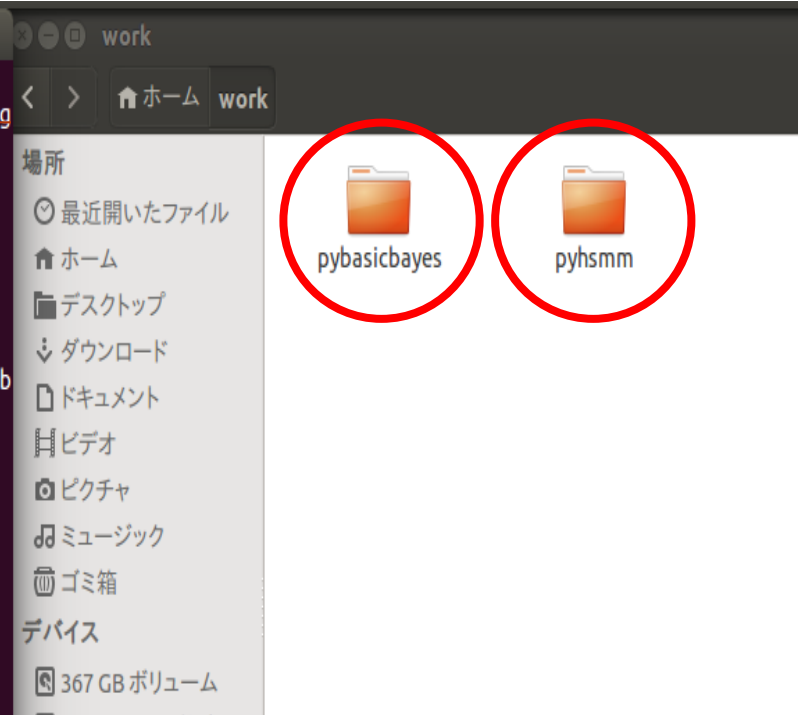
HDP-HSMMライブラリ `pyhsmm` を動かす

Step1 作業フォルダの確保



Step2 必要なライブラリの確保

```
shinshi@TP-T530-2: ~/work
shinshi@TP-T530-2:~$ cd ~/work
shinshi@TP-T530-2:~/work$ git clone --recursive git://github.com/mattjj/pyhsmm.git
Cloning into 'pyhsmm'...
remote: Counting objects: 7616, done.
remote: Total 7616 (delta 0), reused 0 (delta 0), pack-reused 7616
Receiving objects: 100% (7616/7616), 4.05 MiB | 768.00 KiB/s, done.
Resolving deltas: 100% (4028/4028), done.
Checking connectivity... done.
shinshi@TP-T530-2:~/work$ git clone --recursive git://github.com/mattjj/pybasicbayes.git
Cloning into 'pybasicbayes'...
remote: Counting objects: 2142, done.
remote: Total 2142 (delta 0), reused 0 (delta 0), pack-reused 2142
Receiving objects: 100% (2142/2142), 679.89 KiB | 445.00 KiB/s, done.
Resolving deltas: 100% (1323/1323), done.
Checking connectivity... done.
shinshi@TP-T530-2:~/work$
```



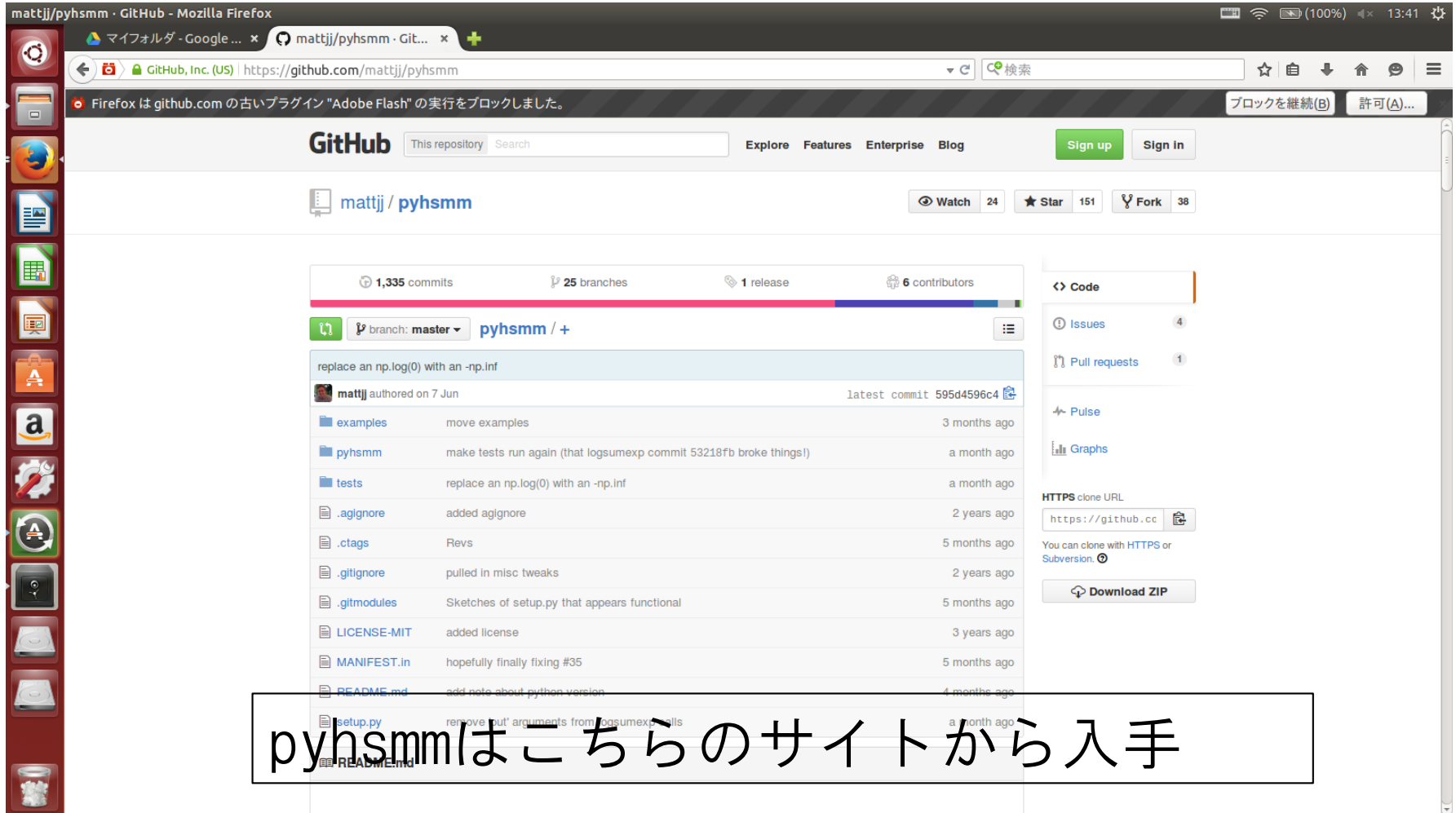
作業フォルダに移動し、コマンド

```
git clone --recursive git://github.com/mattjj/pyhsmm.git
```

```
git clone --recursive git://github.com/mattjj/pybasicbayes.git
```

よりpybasicbayesとpyhsmmを入手する

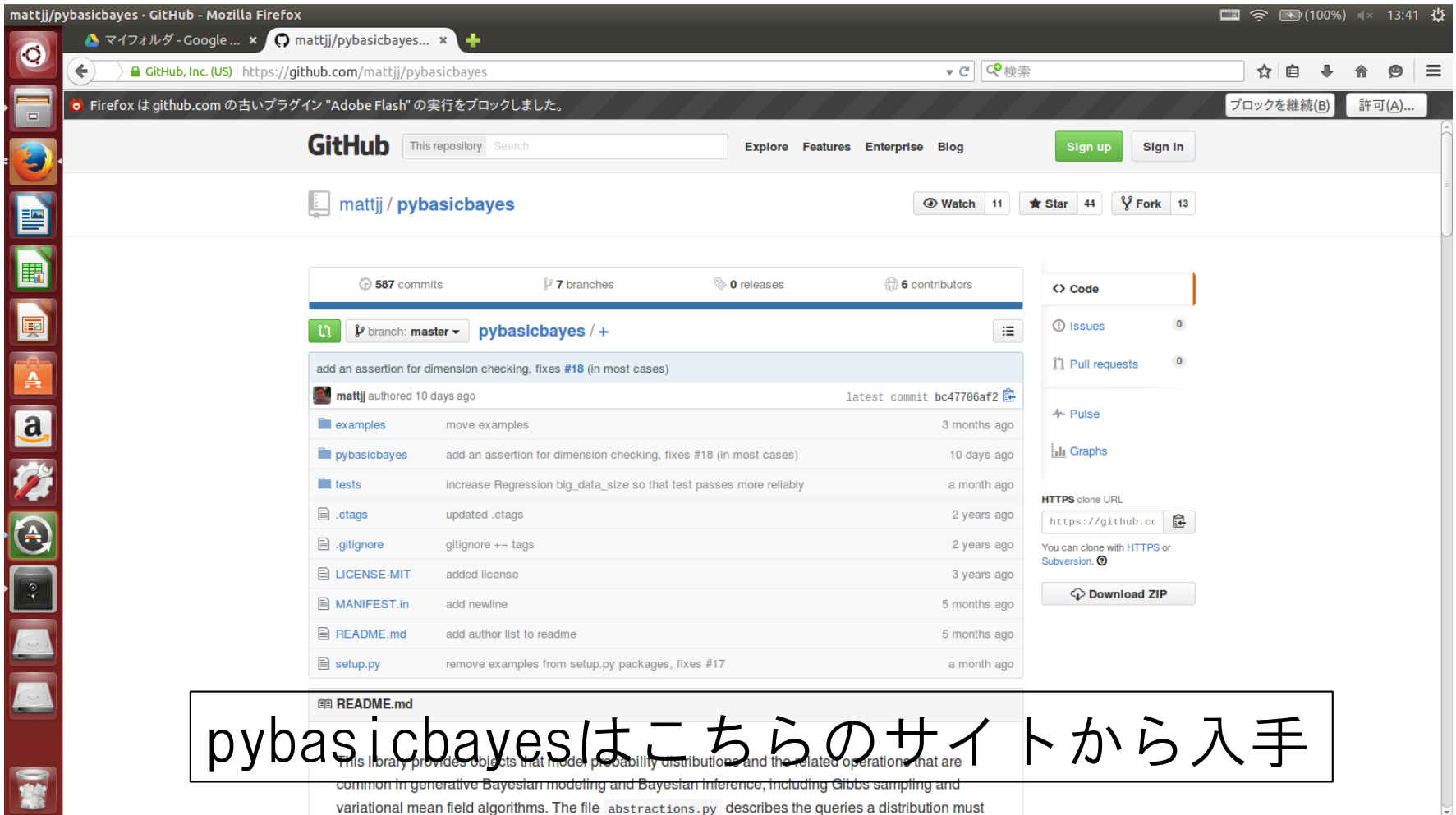
Step2 必要なライブラリの確保



The screenshot shows the GitHub repository page for `mattjj/pyhsmm`. The repository has 1,335 commits, 25 branches, 1 release, and 6 contributors. The current branch is `master`. The commit history shows a recent commit by `mattjj` on 7 Jun, with the message "replace an np.log(0) with an -np.inf". The file list includes `examples`, `pyhsmm`, `tests`, `.agignore`, `.ctags`, `.gitignore`, `.gitmodules`, `LICENSE-MIT`, `MANIFEST.in`, `README.md`, and `setup.py`.

pyhsmmはこちらのサイトから入手

Step2 必要なライブラリの確保



The screenshot shows the GitHub repository page for `mattjj/pybasicbayes`. The repository has 587 commits, 7 branches, 0 releases, and 6 contributors. The latest commit is `bc47706af2` by `mattjj` 10 days ago. The commit message is "add an assertion for dimension checking, fixes #18 (in most cases)". The repository contains the following files:

File	Commit Message	Time Ago
<code>examples</code>	move examples	3 months ago
<code>pybasicbayes</code>	add an assertion for dimension checking, fixes #18 (in most cases)	10 days ago
<code>tests</code>	Increase Regression <code>big_data_size</code> so that test passes more reliably	a month ago
<code>.ctags</code>	updated .ctags	2 years ago
<code>.gitignore</code>	gitignore += tags	2 years ago
<code>LICENSE-MIT</code>	added license	3 years ago
<code>MANIFEST.in</code>	add newline	5 months ago
<code>README.md</code>	add author list to readme	5 months ago
<code>setup.py</code>	remove examples from setup.py packages, fixes #17	a month ago

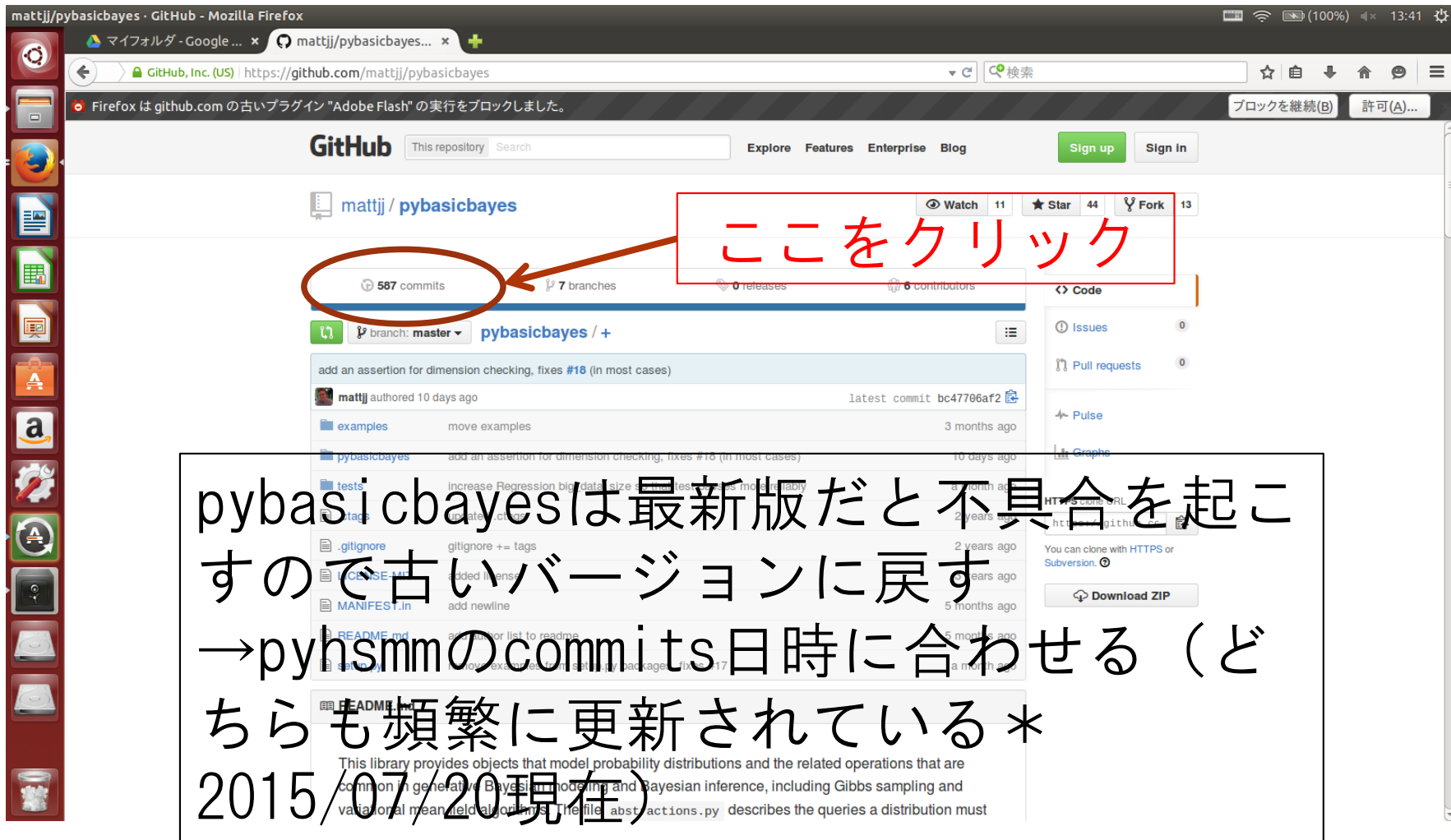
The `README.md` file is highlighted, showing the following text:

```
This library provides objects that model probability distributions and the related operations that are common in generative Bayesian modeling and Bayesian inference, including Gibbs sampling and variational mean field algorithms. The file abstractions.py describes the queries a distribution must
```

A text box at the bottom of the screenshot contains the text:

pybasicbayesはこちらのサイトから入手

Step3 ライブラリのバージョン変更



587 commits

ここをクリック

pybasicbayesは最新版だと不具合を起こすので古いバージョンに戻す
→pyhsmmのcommits日時に合わせる（どちらも頻繁に更新されている＊
2015/07/20現在）

Step3 ライブラリのバージョン変更

ここをクリック
(この日時以降ファイル内容が
変更となりpyhsmmでインポート
しなければならないものがなくな
っている)

add an assertion for dimension checking, fixes #18 (in most cases)
mattjj authored 10 days ago

Commits on Jul 2, 2015

Merge branch 'reorganize-into-files'
mattjj authored 17 days ago

Commits on Jun 7, 2015

increase Regression big_data_size so that test passes more reliably
mattjj authored on 7 Jun

only import matplotlib when necessary
mattjj authored on 7 Jun

move content of labels.py into models.py, fix plotting import error so ...
mattjj authored on 7 Jun

reorganize distributions.py into smaller files
mattjj authored on 7 Jun

remove examples from setup.py packages, fixes #17
mattjj authored on 7 Jun

Commits on May 24, 2015

put back #16 with tweaked logic (new test passes)
mattjj authored on 24 May


move tests, add test for Gaussian() no-arg instantiation (c.f. #16)
mattjj authored on 24 May

revert #16
mattjj authored on 24 May

add back in more tests


Step3 ライブラリのバージョン変更

GitHub [Explore](#) [Features](#) [Enterprise](#) [Blog](#) [Sign up](#) [Sign in](#)

 **mattjj / pybasicbayes** Watch 11 Star 44 Fork 13






remove examples from setup.py packages, fixes #17

[Browse files](#)

 **mattjj** authored on 7 Jun 1 parent [9169c54](#) commit [8eea0e2f9d1f58d63362ceb20794df3f80e9ce2a](#)


Showing 1 changed file with 0 additions and 1 deletion.

Unified Split

1      setup.py [View](#)

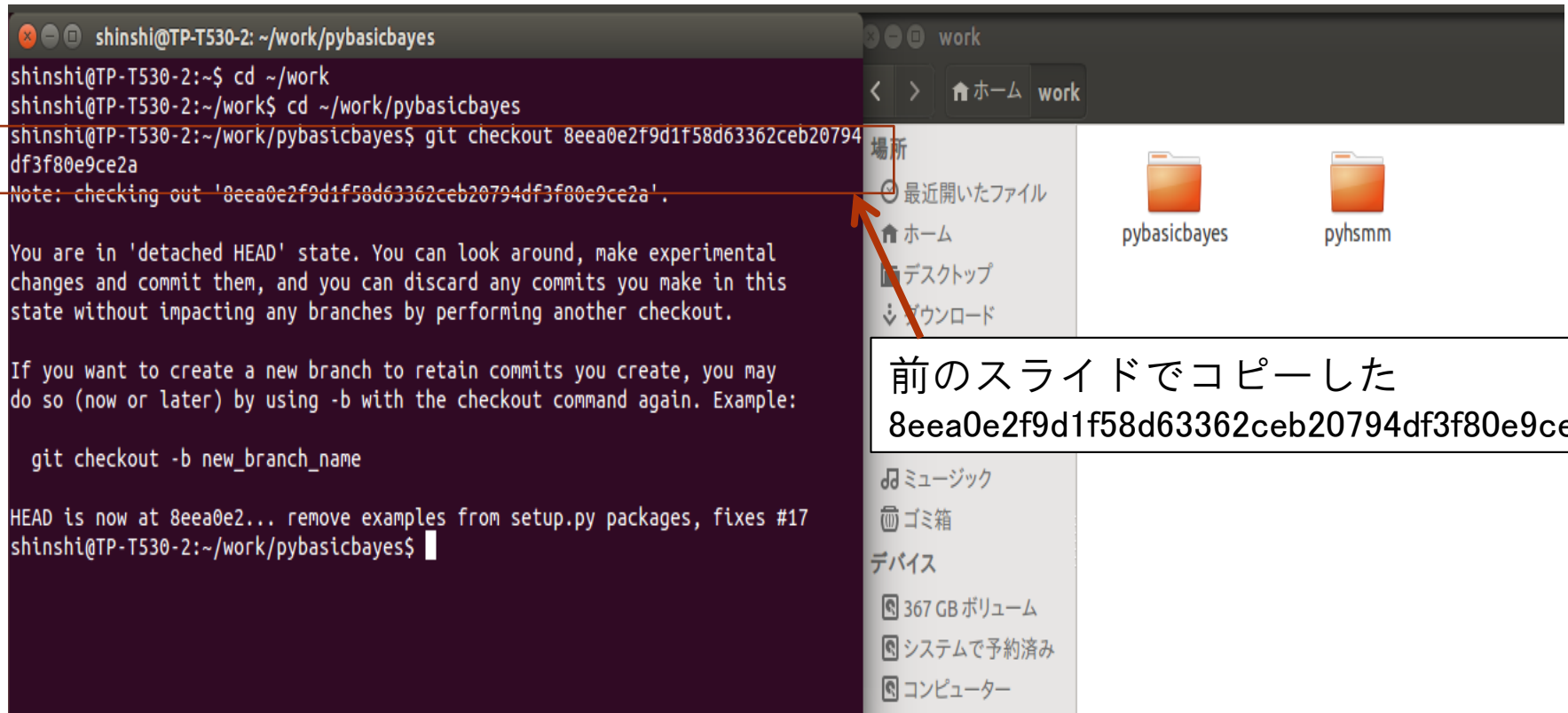
17	17	maintainer_email='mattjj@csail.mit.edu',
18	18	packages=['pybasicbayes',
19	19	'pybasicbayes.internals',
20	-	'pybasicbayes.examples',
21	20	'pybasicbayes.util',
22	21	'pybasicbayes.testing'],
23	22	platforms='ALL',

Please [sign in](#) to comment.

© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#)  [Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#) [Help](#)

この部分をコピー

Step3 ライブラリのバージョン変更



The image shows a terminal window and a file manager side-by-side. The terminal window has a title bar 'shinshi@TP-T530-2: ~/work/pybasicbayes'. The command history shows 'cd ~/work' and 'cd ~/work/pybasicbayes'. The current command is 'git checkout 8eea0e2f9d1f58d63362ceb20794df3f80e9ce2a', which is highlighted with a red box. The output shows a note about checking out the commit and a message about the 'detached HEAD' state. The file manager shows the 'work' directory with folders 'pybasicbayes' and 'pyhsmm'. A red arrow points from the commit hash in the terminal to a text box on the right.

```
shinshi@TP-T530-2: ~/work/pybasicbayes
shinshi@TP-T530-2:~/work$ cd ~/work
shinshi@TP-T530-2:~/work$ cd ~/work/pybasicbayes
shinshi@TP-T530-2:~/work/pybasicbayes$ git checkout 8eea0e2f9d1f58d63362ceb20794df3f80e9ce2a
Note: checking out '8eea0e2f9d1f58d63362ceb20794df3f80e9ce2a'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 8eea0e2... remove examples from setup.py packages, fixes #17
shinshi@TP-T530-2:~/work/pybasicbayes$
```

場所
最近開いたファイル
ホーム
デスクトップ
ダウンロード

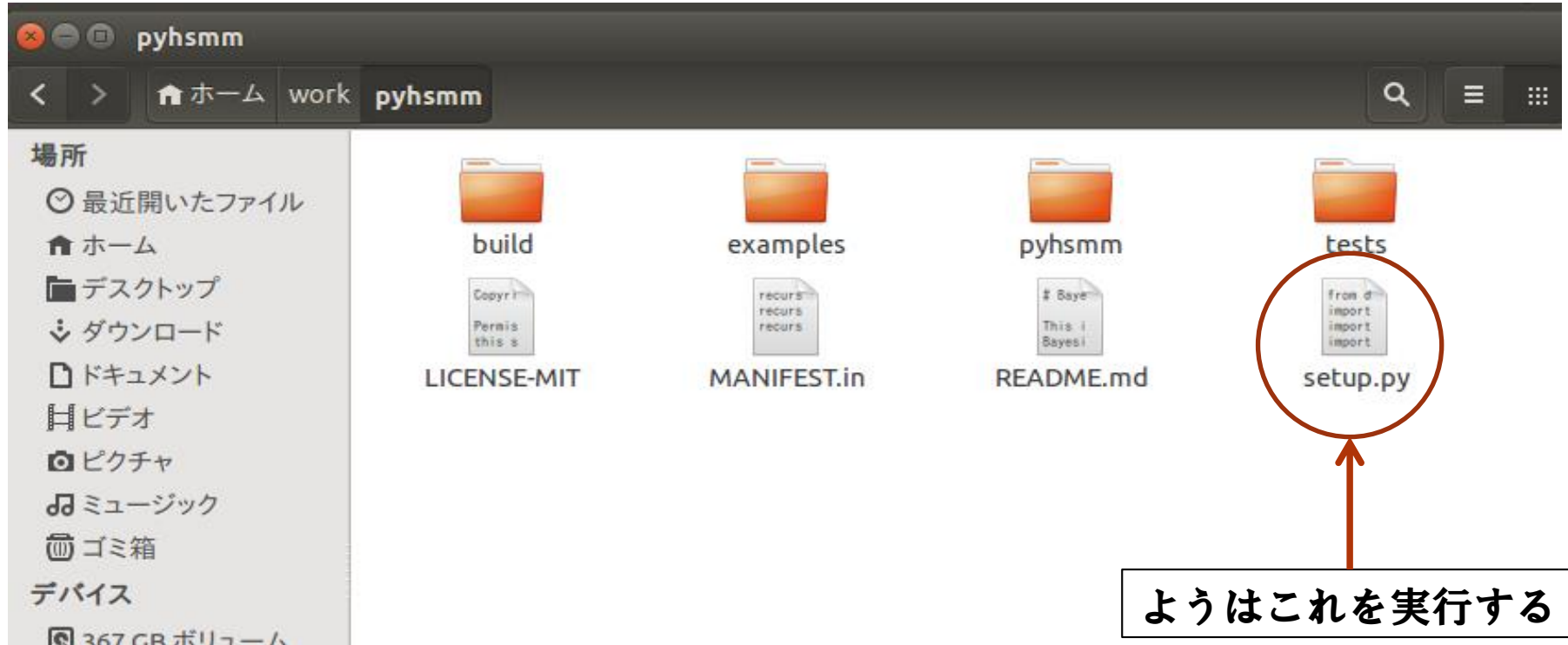
pybasicbayes pyhsmm

前のスライドでコピーした
8eea0e2f9d1f58d63362ceb20794df3f80e9ce2a

ミュージック
ゴミ箱
デバイス
367 GB ボリューム
システムで予約済み
コンピューター

まずpybasicbayesのフォルダ内に移動
次にコマンドgit checkout 前のスライドでコピーしたcommit
よりバージョンの変更を行う

Step4 pyhsmmのコンパイル



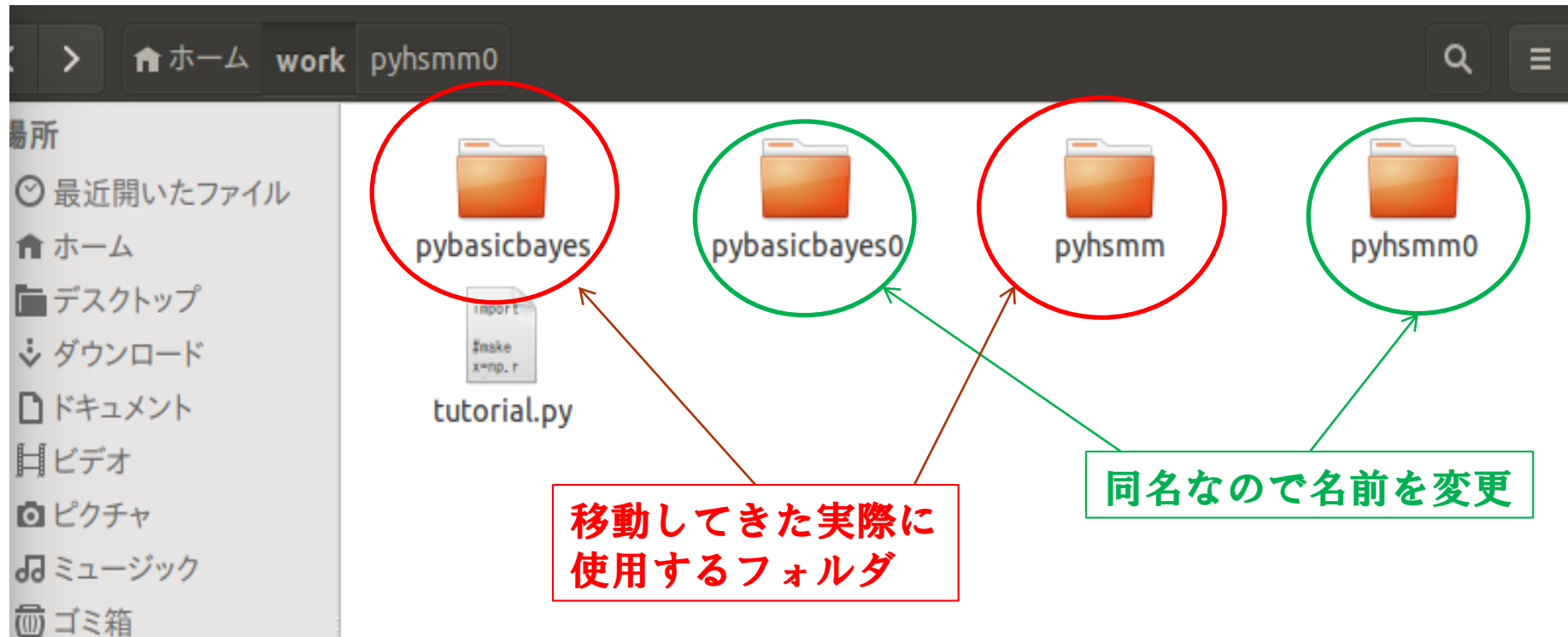
pyhsmm内に移動し、コマンド
`python setup.py build ext --inplace --with-cython`
よりコンパイルを行う（しなくてもよいかも．．．）

Step5 フォルダの移動



pyhsmm内に存在するpyhsmmフォルダを実際に作業で使用するフォルダ内に移動する（今回はworkフォルダ）

Step5 フォルダの移動



pybasicbayesも同様に実際に作業で使用する
フォルダ内に移動する

Step5.5 チュートリアルの実行

- pyhsmmをPython上でインポートするには作業フォルダ（今回はwork）にパスを通す必要がある
- シェル開始されるたび（ターミナルを開くたび）にパスを通したいので、シェル開始時設定ファイル”.*rc（*:bash, zsh, shなど）”に変更を加える
 - `export PYTHONPATH=/home/work`を書き加える

パスの通し方は他にもいろいろあります

Step6 チュートリアルの実行



pyhsmmとpybasicbayesが正しくインストール出来ているか
確かめるためにtutorialを実行する

Step6 チュートリアルの実行

```
shinshi@TP-T530-2: ~/work
ges/numpy/core/include -I/usr/include/python2.7 -c pyhsmm/internals/hmm_messages
_interface.cpp -o build/temp.linux-x86_64-2.7/pyhsmm/internals/hmm_messages_inte
rface.o -std=c++11 -O3 -w -DNDEBUG -DHMM_TEMPS_ON_HEAP
cc1plus: warning: command line option '-Wstrict-prototypes' is valid for C/Objc
but not for C++ [enabled by default]
c++ -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-Bsymbolic-functions -
Wl,-z,relro -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototy
pes -D_FORTIFY_SOURCE=2 -g -fstack-protector --param=ssp-buffer-size=4 -Wformat -
Werror=format-security build/temp.linux-x86_64-2.7/pyhsmm/internals/hmm_messages
_interface.o -o /home/shinshi/work/pyhsmm/pyhsmm/internals/hmm_messages_interfac
e.so
shinshi@TP-T530-2:~/work/pyhsmm$ cd ~/work
shinshi@TP-T530-2:~/work$ python tutorial.py
/home/shinshi/work/pybasicbayes/distributions.py:27: UserWarning: using slow sam
ple_crp_tablecounts
warn('using slow sample_crp_tablecounts')
..... [ 25/100, 0.02sec avg, ETA 1.13 ]
..... [ 50/100, 0.02sec avg, ETA 0.75 ]
..... [ 75/100, 0.01sec avg, ETA 0.37 ]
..... [ 100/100, 0.01sec avg, ETA 0.00 ]

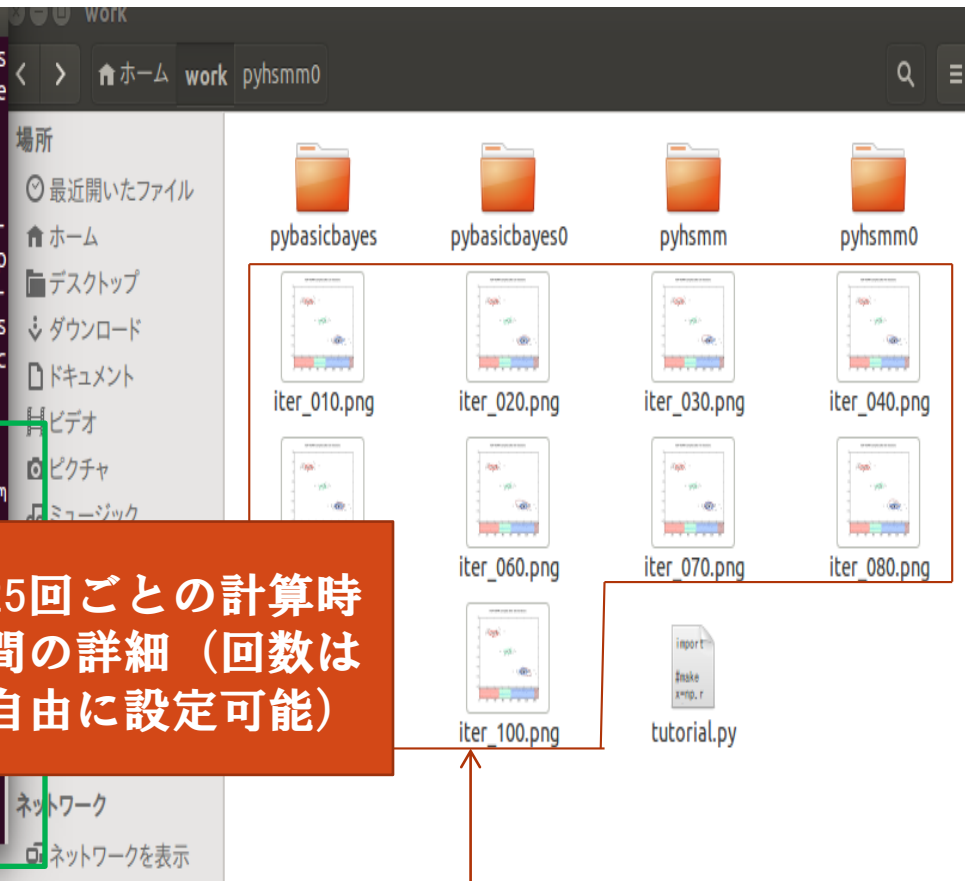
0.01sec avg, 1.50 total
shinshi@TP-T530-2:~/work$
```

25回ごとの計算時
間の詳細（回数は
自由に設定可能）

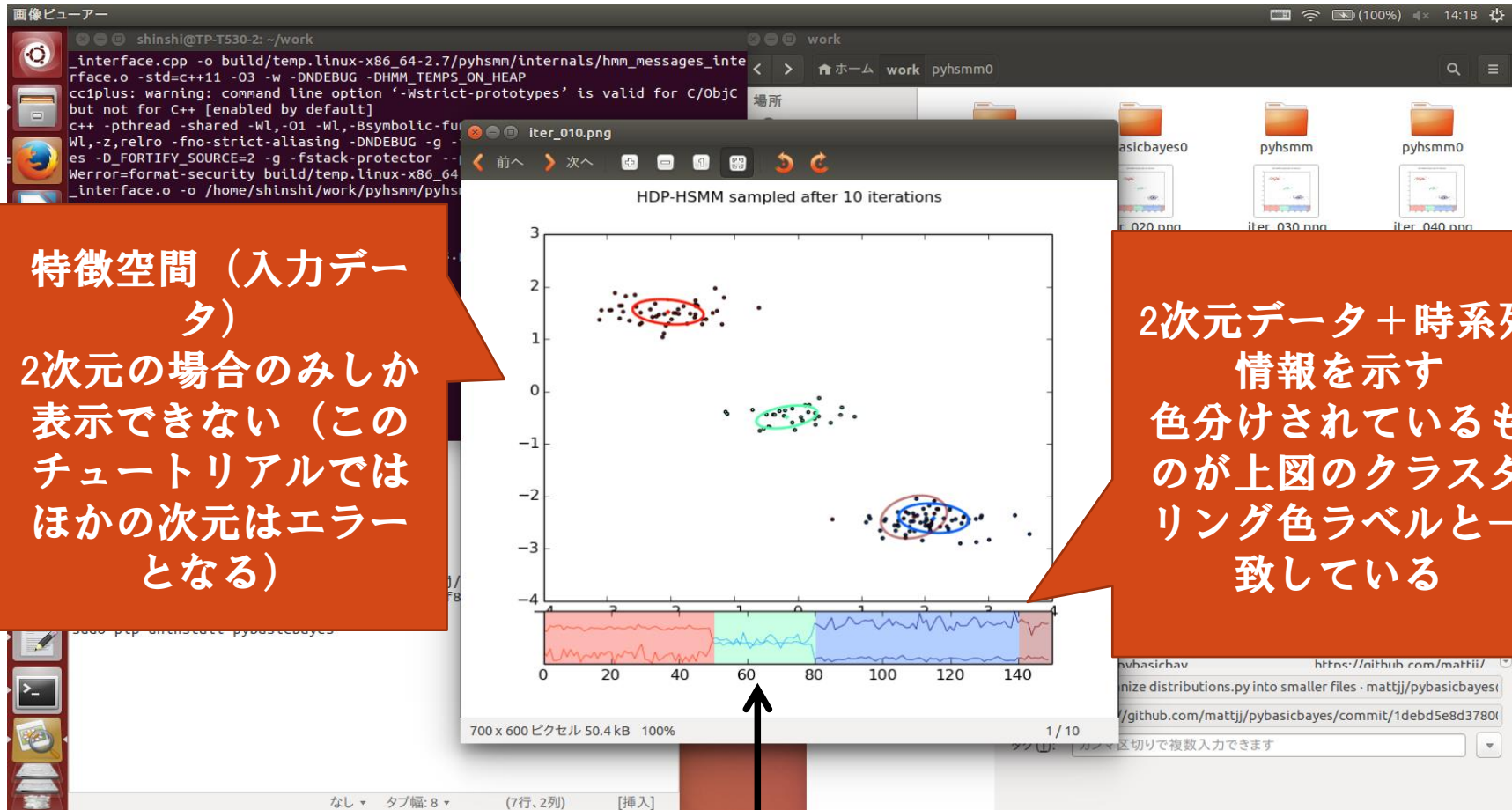
合計時間

tutorialを実行した時の端末の状況

このようなファイルで出来ていると成功



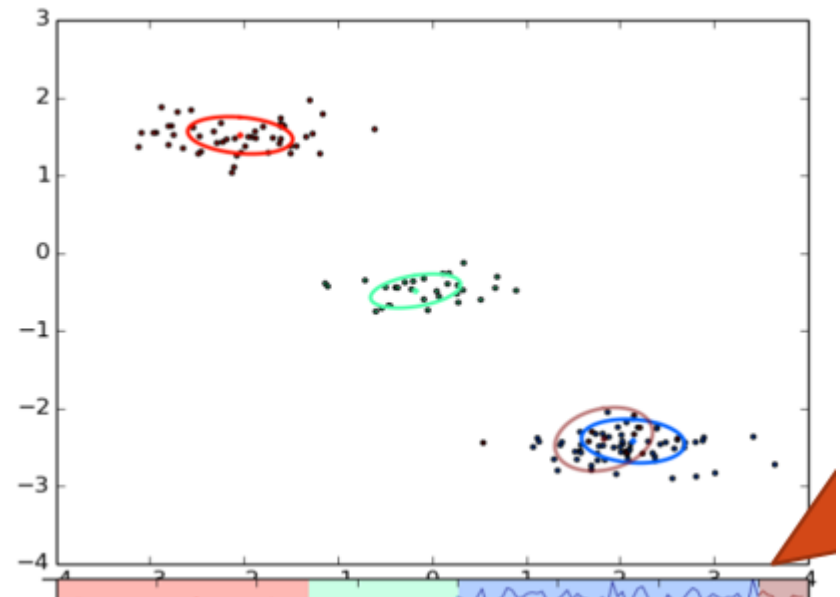
Step6 チュートリアルの実行



Step6.5 チュートリアルプログラム詳細

```
tutorial.py x
1 import numpy as np
2
3 #make sample data
4 x=np.random.normal(-2,0.5,50)
5 x2=np.random.normal(0,0.5,30)
6 x3=np.random.normal(2,0.5,70)
7 y=np.random.normal(1.5,0.2,50)
8 y2=np.random.normal(-0.5,0.2,30)
9 y3=np.random.normal(-2.5,0.2,70)
10 X=x.tolist()+x2.tolist()+x3.tolist()
11 Y=y.tolist()+y2.tolist()+y3.tolist()
12 data = zip(X,Y)
```

- 時系列データクラスタリングのためのサンプルデータを作成
 - 2次元（以下みたいなことができる）



Step6.5 チュートリアルプログラム詳細

```
17 import pyhsmm
18 import pyhsmm.basic.distributions as distributions
19
20 obs_dim = 2
21 Nmax = 10
22
23 obs_hypparams = {'mu_0': np.zeros(obs_dim),
24                  'sigma_0': np.eye(obs_dim),
25                  'kappa_0': 0.3,
26                  'nu_0': obs_dim+5}
27 dur_hypparams = {'alpha_0': 500,
28                  'beta_0': 10}
29
30 obs_distns = [distributions.Gaussian(**obs_hypparams) for state in range(Nmax)]
31 dur_distns = [distributions.PoissonDuration(**dur_hypparams) for state in range(Nmax)]
32
33 posteriormodel = pyhsmm.models.WeakLimitHDPHSM(
34     alpha=6., gamma=6., # better to sample over these; see concentration-resampling.py
35     init_state_concentration=6., # pretty inconsequential
36     obs_distns=obs_distns,
37     dur_distns=dur_distns)
```

今回用いるHDP-HSMMモデルのオブジェクトを生成（ここではWeakLimitHDPHSM）

- obs_distns, dur_distnsが引数として渡されている

ハイパーパラメータの設定は適切に行う必要がある
（とくにポアソン分布のパラメータ設定は重要）

- pyhsmmをインポート
- 次元数と最大打ち切り状態数設定（計算軽減のため、理論的には ∞ ）
- 仮定する観測分布のパラメータ設定（ガウス分布を想定）
- 仮定する状態の持続時間の分布パラメータ設定（ポアソン分布を想定）

Step6.5 チュートリアルプログラム詳細

```
42 posteriormodel.add_data(data, trunc=60)
```

- 入力データをモデルのオブジェクトに設定
 - 先程作成したサンプルデータを追加する
 - 複数データを追加することで並列に推定アルゴリズムを処理することが可能（詳細は未調査）
- truncで状態の最大持続時間の打ち切り時間を設定可能
 - バックワードメッセージの計算量削減のため

この時点で入力データに関する状態遷移確率などの計算がされる？
（要調査）

Step6.5 チュートリアルプログラム詳細

```
49 from pyhsmm.util.text import progprint_xrange
50 import copy
51
52 models = []
53 ▼ for idx in progprint_xrange(100):
54     ... posteriormodel.resample_model()
55     ... if (idx+1) % 10 == 0:
56         ... models.append(copy.deepcopy(posteriormodel))
```

```
def progprint(iterator, total=None, perline=25, show_times=True):
    times = []
    idx = 0
    if total is not None:
        numdigits = len('%d' % total)
        for thing in iterator:
            prev_time = time.time()
            yield thing
            times.append(time.time() - prev_time)
            sys.stdout.write('.')
            if (idx+1) % perline == 0:
                if show_times:
                    avgttime = np.mean(times)
                    if total is not None:
                        eta = sec2str(avgttime*(total-(idx+1)))
                        sys.stdout.write(' [ %dd/%dd, %7.2fsec avg, ETA %s ]\n'
                                         % (numdigits, numdigits)) % (idx+1, total, avgttime, eta))
                    else:
                        sys.stdout.write(' [ %d done, %7.2fsec avg ]\n' % (idx+1, avgttime))
                else:
                    if total is not None:
                        sys.stdout.write(' [ %dd/%dd ]\n' % (numdigits, numdigits)) % (idx+1, total))
                    else:
                        sys.stdout.write(' [ %d ]\n' % (idx+1))
            idx += 1
        sys.stdout.flush()
    print ''
    if show_times and len(times) > 0:
        total = sec2str(seconds=np.sum(times))
        print '%7.2fsec avg, %s total\n' % (np.mean(times), total)
```

- ギブスサンプリングによる推定部分（今回は100イテレーション
 - 10回ごとに結果のオブジェクトを保存
 - copy.deepcopy: オブジェクトをコピーするライブラリ

要約：25イテレーションごと（変更可）に計算結果を表示，最後にトータル時間を表示する関数

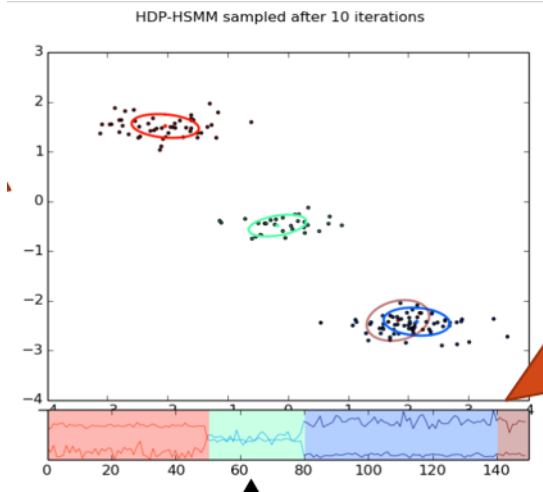
Step6.5 チュートリアルプログラム詳細

```
61 #plot samples
62 from matplotlib import pyplot as plt
63
64 fig = plt.figure()
65 for idx, model in enumerate(models):
66     plt.clf()
67     model.plot()
68     plt.gcf().suptitle('HDP-HSMM sampled after %d iterations'%(10*(idx+1)))
69     plt.savefig('iter_%.3d.png'%(10*(idx+1)))
70
71 #dump object
72 """
73 models = [posteriormodel.resample_and_copy() for itr in progprint_xrange(150)]
74
75 import cPickle
76 with open('sampled_models.pickle','w') as outfile:
77     cPickle.dump(models,outfile,protocol=-1)
78
79 #load pickle file
80 with open("sampled_models.pickle", 'rb') as pickle_file:
81     models = cPickle.load(pickle_file)
82 """
```

コメントアウト部分：

- サンプルング部分はリスト内包表記で簡略化可能
- Pickle (cPickle) を用いることで結果のオブジェクトをファイル保存、読み込みできる

- 部分で保存したオブジェクト
(ここでは10個) に関する推定結果を以下のようにプロットしている



次回

- NPB-DAA（ノンパラメトリック二重分節解析器）を動かす
 - pyhsmmを継承して作成されている
 - 入力データは日本語話者による母音列のみで構成された音声データ
 - 音声データ→MFCC12次元→DSAE3次元
 - summary.pyは結果の評価，図示プログラム
 - VMWARE＋Ubuntu64bit_ver12.04（surface pro3, windows8上）で動作確認済み
 - 計算量が膨大なためハイスペックな（いろいろな意味で）PCが好ましい