Python 3

Neal Morton

Python changes

- Python language developers decided that there needed to be a set of backwards incompatible changes
    - Python 3 released on 2008-12-03
    - Python 2 support extended to 2020-01-01
    - Only bug fixes after that
- Requires changes to most scripts

Overview

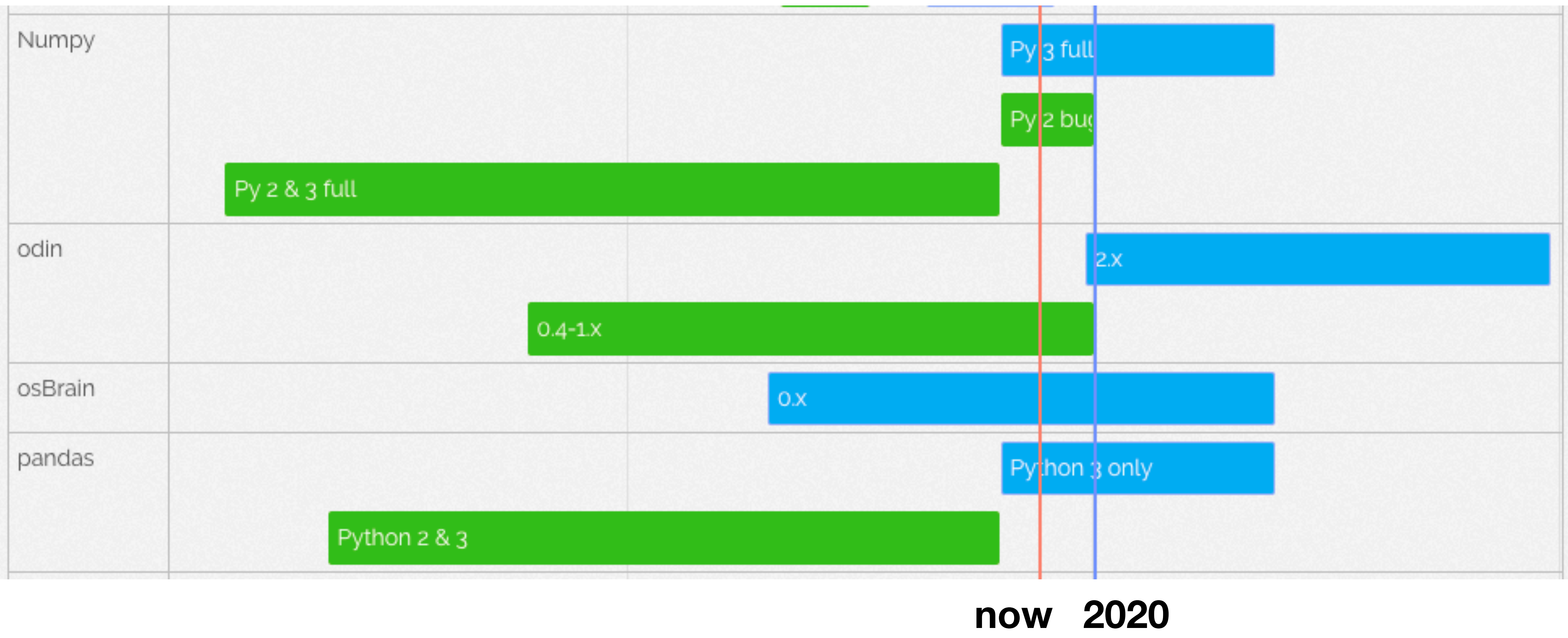
- Changes between python 2 and 3
- Making code compatible with python 3
- Effects on python packages we use
- Demo of a searchlight analysis using python 3

Some relevant changes

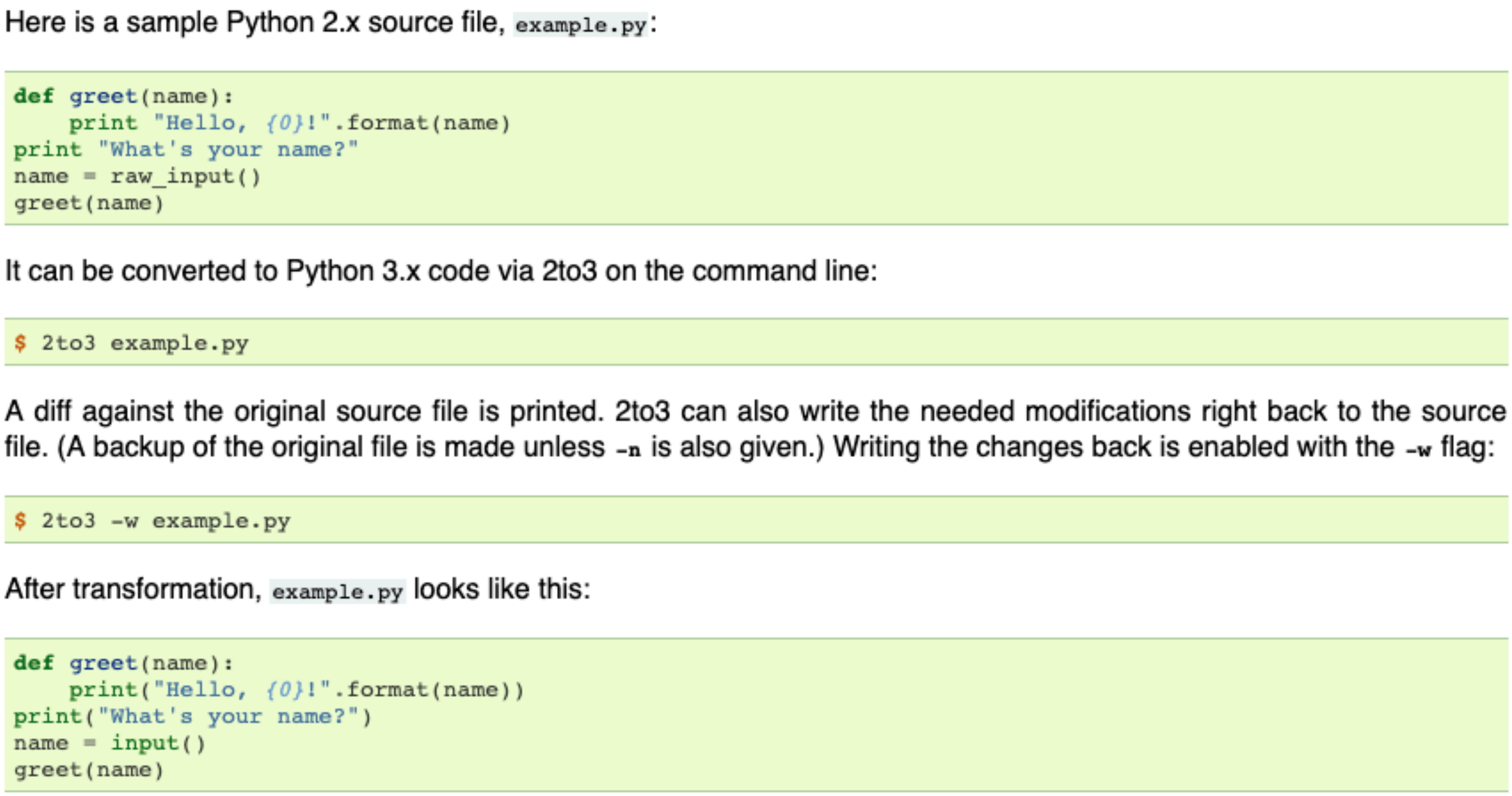
- `print x # python 2`
- `print(x) # python 3`
- `3/2 # in python 2, gives 1; in python 3, is 1.5`
- Some changes in how dict objects work to make looping over values in a dict more efficient

    -
      ```
      d = {'a':1, 3:4}
      for key, value in d.items():
          print('key: {}, value: {}'.format(key, value)
      ```
- For some special cases, need the "six" module to write code that is compatible with both python 2 and 3

Python 2 sunsetting



Automated code conversion



Here is a sample Python 2.x source file, `example.py`:

```
def greet(name):
    print "Hello, {0}!".format(name)
print "What's your name?"
name = raw_input()
greet(name)
```

It can be converted to Python 3.x code via 2to3 on the command line:

```
$ 2to3 example.py
```

A diff against the original source file is printed. 2to3 can also write the needed modifications right back to the source file. (A backup of the original file is made unless `-n` is also given.) Writing the changes back is enabled with the `-w` flag:

```
$ 2to3 -w example.py
```

After transformation, `example.py` looks like this:

```
def greet(name):
    print("Hello, {0}!".format(name))
print("What's your name?")
name = input()
greet(name)
```

Python on TACC

- Python packages can be installed in three main places:
  - /work/IRC/ls5/opt – BIC-supported software (Chad Cumba)
  - /work/IRC/ls5/opt/local – user-supported software
  - Your home directory
- module load python3
- module use /work/IRC/ls5/opt/local/modules; module load neuropy3

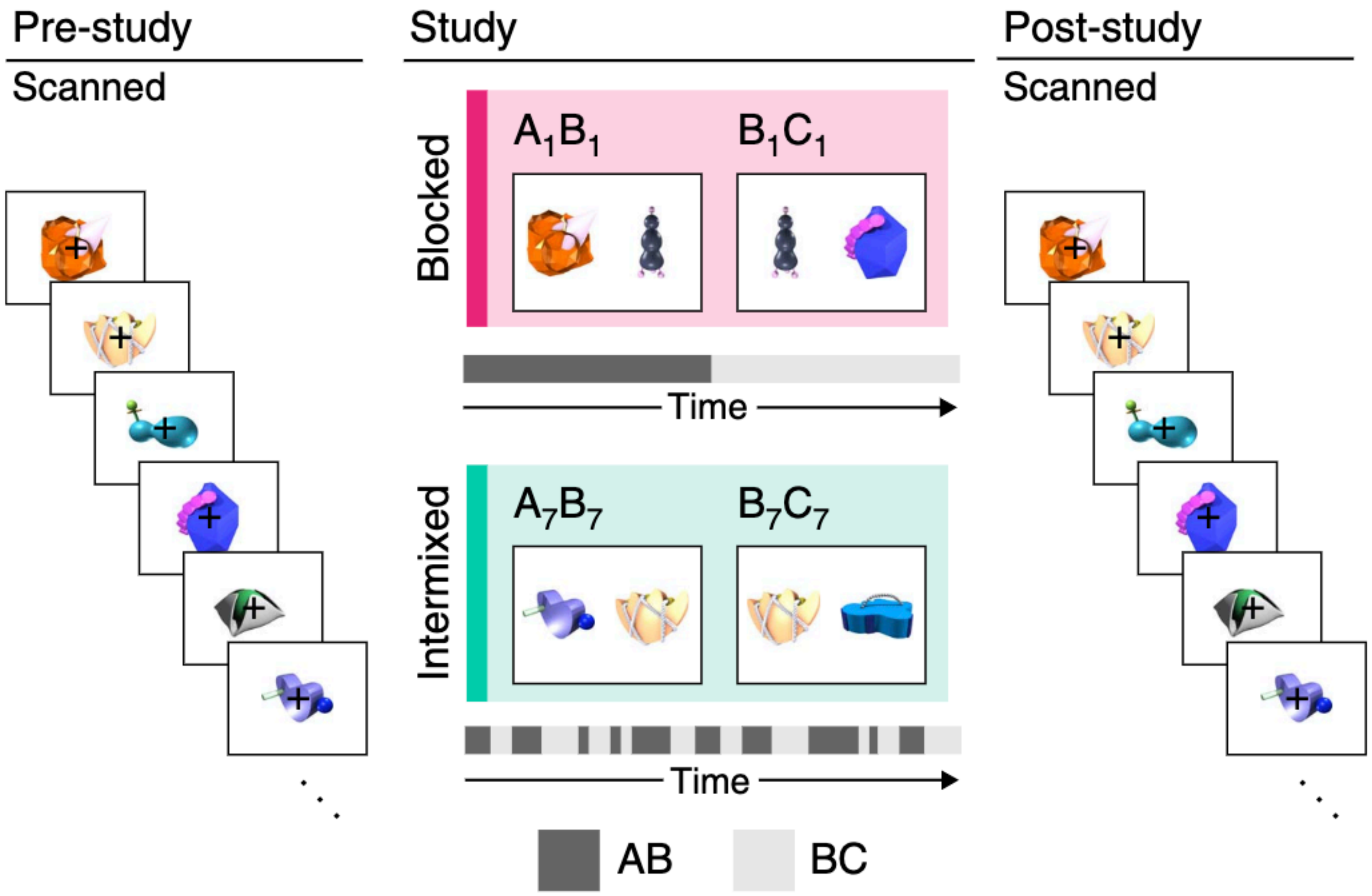
pyMVPA

- Used extensively in the lab for loading datasets in python, running pattern classification and searchlight analyses
- Limited python 3 compatibility (relies on automated fixes)
- Last official release was on 2018-06-26 (more recent changes on GitHub though)
- Includes some packages that are not compatible with python 3
- Seems to work on python 3, but with a lot of warnings

pyMVPA alternatives

- nilearn
    - wide range of pattern classifiers available (everything in scikit-learn)
    - searchlight support
    - design is less flexible than pyMVPA
    - part of the larger nipy ecosystem (e.g. nipype, nibabel)
- brainiak
    - developed by researchers at Princeton
    - designed for advanced analyses like FCMA

Searchlight demo



- Representational similarity analysis in a searchlight
- Test for representations of different task features
- Uses partial RSA from mindstorm

Recommendations

- When starting a new project, use python 3
- For new projects involving multivariate analysis, consider using nilearn or brainiak instead of pymvpa
- It's not too difficult to both use python 3 for new projects and python 2 for old projects on TACC

  -
    ```
    module load python2
    module load ircpy
    module load neuropy
    ```

  -
    ```
    module load python3
    module load neuropy3
    ```
- Make the version of python explicit in scripts (python may be interpreted as python2, not python3)
  - `#!/usr/bin/env python2`
  - `#!/usr/bin/env python3`

Development

- FAT has been converted with the 2to3 utility, attempting to be compatible with python 2 and 3
  - not everything is tested; there may be issues or code that still needs work to be compatible with python 3
- Study-specific python code will need to be converted
- Wiki page for recommendations and issues:
  - https://github.com/prestonlab/wiki/wiki/Python3