ScannerBit

Workgroup update

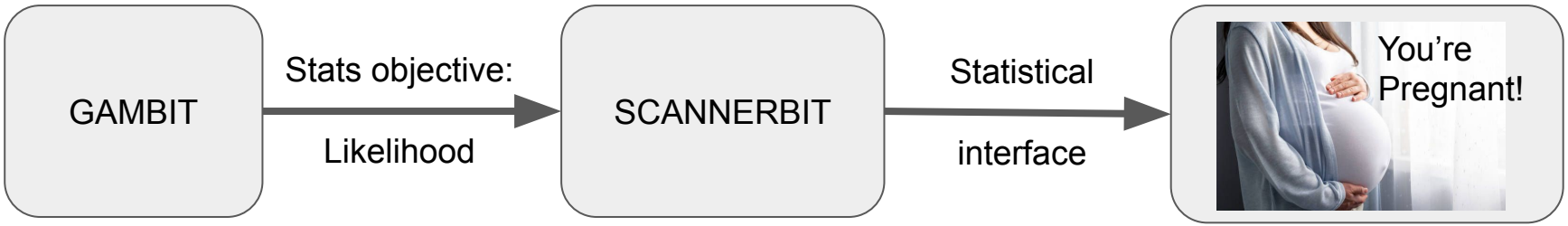
Gregory Martinez, *Andrew Fowlie*, Anders Kvellested, Will Handley

What the F*&K is GAMBIT?

What the F*&K is GAMBIT?
Who the F*&K cares!

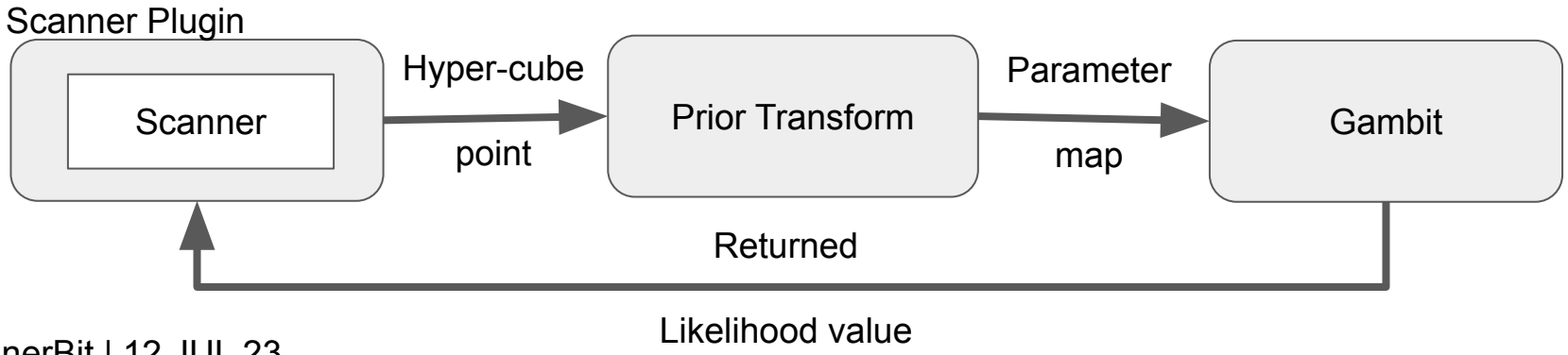
What it ScannerBit?

- Gambit's statistical inference bit
- Four active members currently – More are definitely welcome.
- Perfect for anyone who doesn't care about "science".
- ScannerBit is practically in its own environment where Gambit treated as a black box.



Quick overview

- ScannerBit “scans” in an unit hyper-cube.
- Each scanned hyper-cube point is converted to parameter points via the prior probability density
- Each scanner is contained within a self-contained plugin that is compiled separately from Gambit.



Python plugin

- Python plugin nearly complete. Only a few things to finalize, but should be pushed into master this week.
- End user interface unchanged – Works like any other scanner.
- Diagnostics uses the doc strings – no inifile entry needed.
- Andrew worked a lot on making the interface as friendly to Python programmers as possible.
  - Utils for version, doc string copying and mpi interface
  - Convenient base class
- See ScannerBit/src/scanners/python/plugins/grid.py
- … or anything in ScannerBit/src/scanners/python/plugins
- https://gambit.hepforge.org/downloads/?f=GAMBIT/gambit-2.5.0_beta_python_scanners.tar.gz

Quick Python Plugin Tutorial

- See ScannerBit/src/scanners/python/plugins/grid.py
- … or anything in ScannerBit/src/scanners/python/plugins
- Tarball: https://gambit.hepforge.org/downloads/?f=GAMBIT/gambit-2.5.0_beta_python_scanners.tar.gz – about a 50% it'll work.

Python Plugin Example

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

Python Plugin Example

Pugin Interface:
- __init__(**args)
- run()
- __version__
- __plugins__

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

Python Plugin Example

Pugin Interface:
- __init__(**args)
- run()
- __version__
- __plugins__

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

scanner base class:
- loglike_hypercube
- loglike_physical
- run_args
- dim

Python Plugin Example

Pugin Interface:
- __init__(**args)
- run()
- __version__
- __plugins__

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

Documentation
- Diagnostics will use docs strings
- Will separate __init__ and run doc strings

scanner base class:
- loglike_hypercube
- loglike_physical
- run_args
- dim

Python Plugin Example

Pugin Interface:
- __init__(**args)
- run()
- __version__
- __plugins__

Utils:
- version
- @copydoc
- mpi

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

Documentation
- Diagnostics will use docs strings
- Will separate __init__ and run doc strings

scanner base class:
- loglike_hypercube
- loglike_physical
- run_args
- dim

Python Plugin Example

Pugin Interface:
- __init__(**args)
- run()
- __version__
- __plugins__

```
import scipy.optimize
import scanner_plugin as splug
from utils import copydoc, version

class DifferentialEvolution(splug.scanner):
    """
    Differential evolution optimizer from scipy.
    """

    __version__ = version("scipy")

    def __init__(self, **kwargs):
        super().__init__()

    @copydoc(scipy.optimize.differential_evolution)
    def run(self):
        bounds = [(0., 1.)] * self.dim
        res = scipy.optimize.differential_evolution(
            self.loglike_hypercube, bounds, **self.run_args)
        print(res)

__plugins__ = { “scipy_differential_evolution”: DifferentialEvolution}
```

Documentation
- Diagnostics will use docs strings
- Will separate __init__ and run doc strings

scanner base class:
- loglike_hypercube
- loglike_physical
- run_args
- dim

Utils:
- version
- @copydoc
- mpi

Inifile interface is same

```
Scanner:
  use_scanner: scipy_differential_evolution

  scanners:
    scipy_differential_evolution:
      like: LogLike
      plugin: scipy_differential_evolution
      run:
        strategy: 'best1bin'
        maxiter: 1000
        popsize: 15
        tol: 0.01
        mutation: [0.5, 1]
        recombination: 0.7
        disp: false
        polish: true
        init: 'latinhypercube'
        atol: 0
        updating: 'immediate'
```

Added “pkg” option to force load scanner.

```
Scanner:
  use_scanner: scipy_differential_evolution

  scanners:
    scipy_differential_evolution:
      like: LogLike
      plugin: scipy_differential_evolution
      pkg: gambit_scipy
      run:
        strategy: 'best1bin'
        maxiter: 1000
        popsize: 15
        tol: 0.01
        mutation: [0.5, 1]
        recombination: 0.7
        disp: false
        polish: true
        init: 'latinhypercube'
        atol: 0
        updating: 'immediate'
```

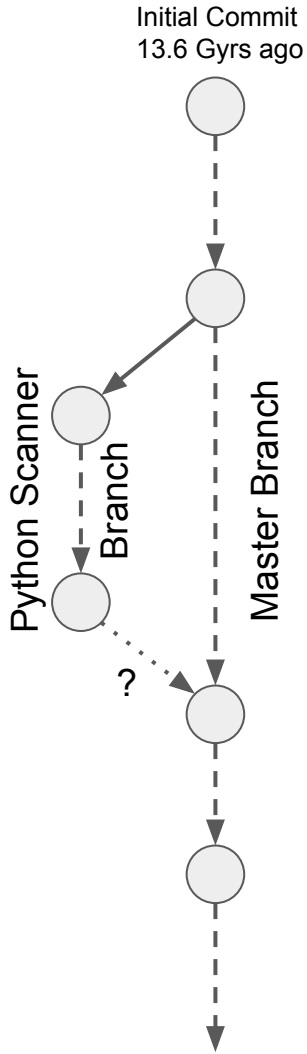
Prior Interface Changes: Conversion to Eigen types

```
class RangePrior1D : public BasePrior
{
    void transform(const std::vector<double> unitpars,
                   std::unordered_map<std::string,double> &output) const
    {...}
};
```

```
class RangePrior1D : public BasePrior
{
    void transform(hyper_cube<double> unitpars,
                   std::unordered_map<std::string,double> &output) const
    {...}
};
```

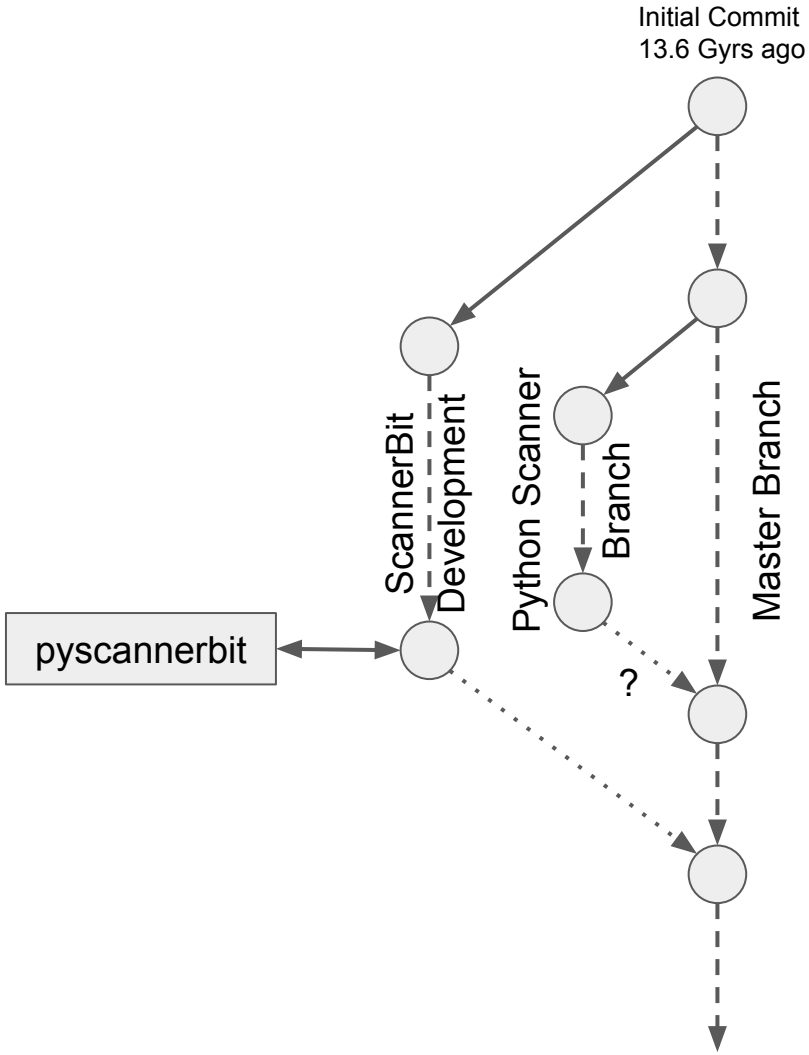
Current and Future Plans

- Merge Python Scanner Branch
  - Issues with base class.
  - Finalize interface.
  - Conversion to Eigen has begun.



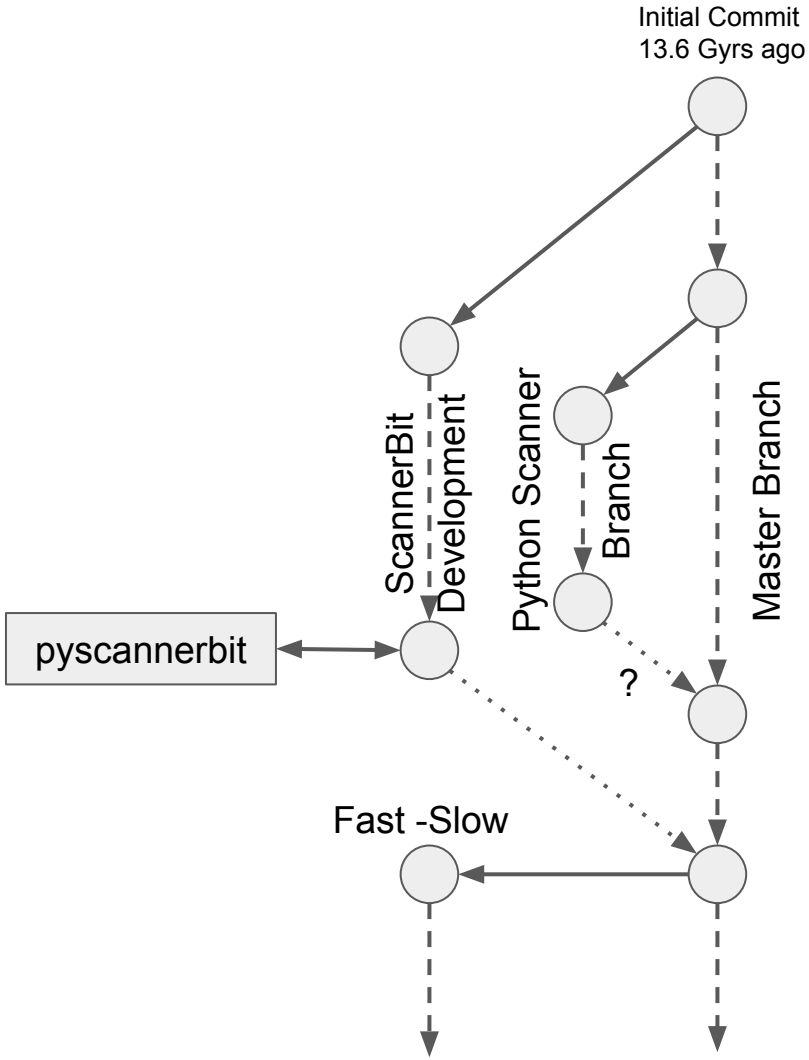
Current and Future Plans

- Merge Python Scanner Branch
  - Issues with base class.
  - Finalize interface.
  - Conversion to Eigen has begun.
- Merge ScannerBit_develoment
  - Has bunch of scanners.
  - Issues since it interfaces with pyscannerbit.



Current and Future Plans

- Merge Python Scanner Branch
  - Issues with base class.
  - Finalize interface.
  - Conversion to Eigen has begun.
- Merge ScannerBit_develoment
  - Has bunch of scanners.
  - Issues since it interfaces with pyscannerbit.
- Fast-Slow development
  - Convert to Eigen vectors.
  - Allow sub-hyper-cube inputs to likelihoods
  - Group parameters.



Discussion and bits

- New python scanners development can start now, I mean right now, like at this moment, don’t waste another minute, code code code!
- Evil setfault bug solved – Python variable called before interpreter starting.
  - Currently fixed by starting interpreter forever, but smarter solution is being devised by Pat.
- Future continuous learning plugin?
  - Framework already exists on ScannerBit side.
- Fast-slow discussion – but later.