Tassel 5 Pipeline Tutorial *(Command Line Interface)*

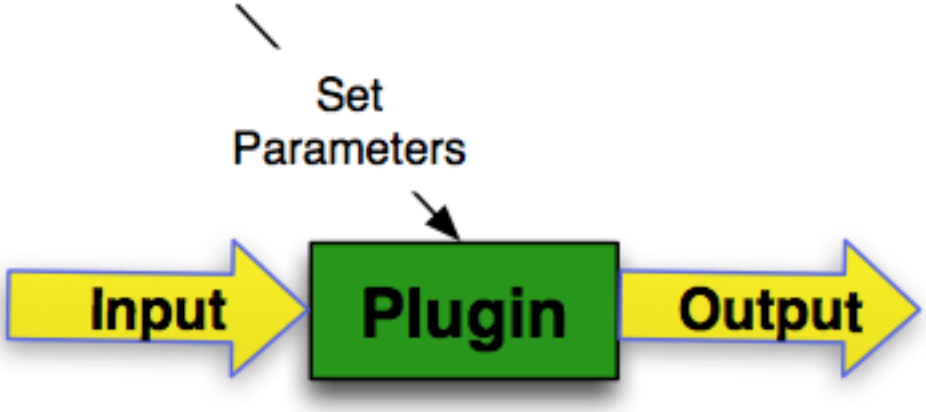
Terry Casstevens

Institute for Genomic Diversity, Cornell University

March 30, 2016

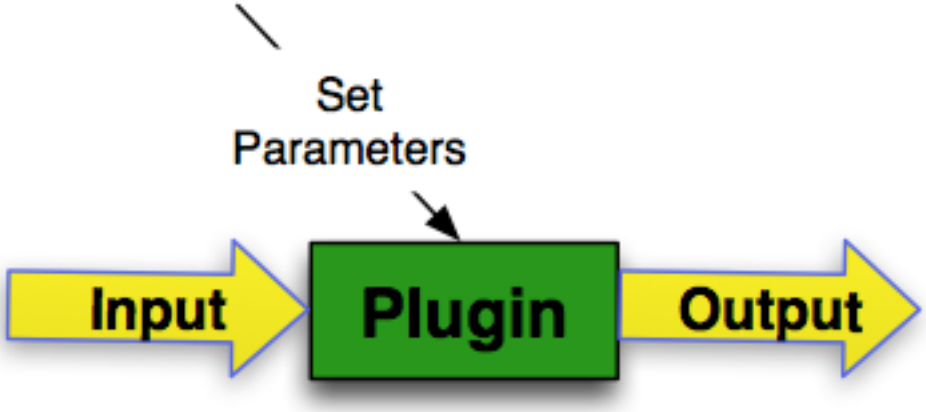
Tassel Plugin…

- Plugins are **Modularized Functionality** (i.e. import, export, GLM, MLM, Kinship, LD, etc.)
- First step is to **set parameters**. Although this could be optional as, defaults may be provided or there are none.
- **Input** may come from the Output of another Plugin.
- The Plugin may produce **Output**.



Setting Plugin Parameters…

- -flag (true or false)
  - -includeSites
  - -includeSites false
- -flag <value>
  - -siteMinAlleleFreq 0.01



Tassel Pipeline…

- **Plugins can be chained together.**

```
./run_pipeline.pl <plugin> <plugin> <plugin>
```

Tassel Plugin Usage…

- **Some are Legacy Flags (i.e. doesn't use -endPlugin)**

https://bitbucket.org/tasseladmin/tassel-5-source/wiki/docs/Tassel5PipelineCLI.pdf

-importGuess

-intersect

- **Others follow our Preferred Plugin Design**

https://bitbucket.org/tasseladmin/tassel-5-source/wiki/docs/TasselSelfDescribingPlugin.pdf

<Plugin Name> <Parameters> -endPlugin

-KinshipPlugin -method Centered_IBS -endPlugin

Tassel Available Plugins…

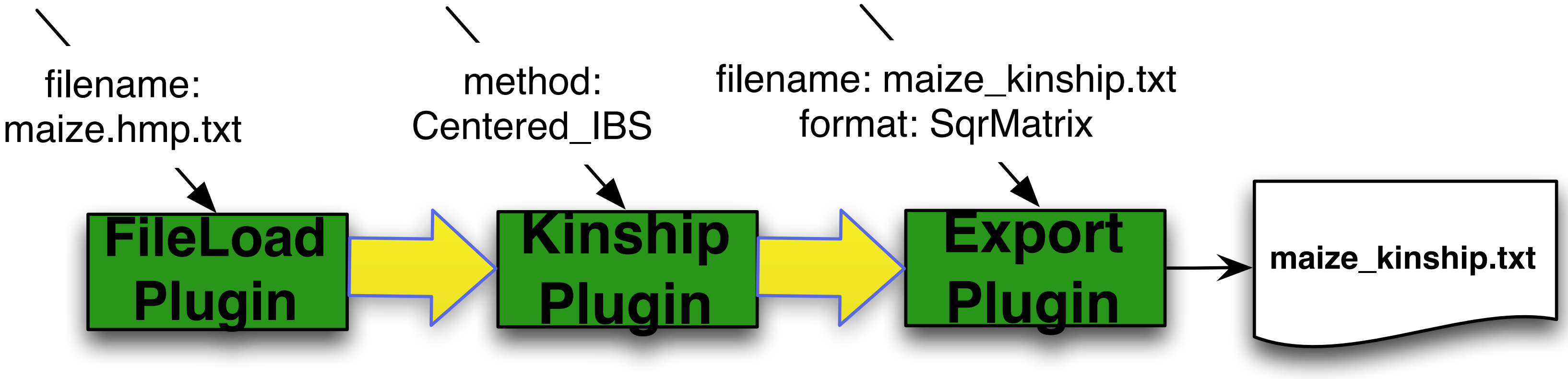
(Plugins following preferred design)

```
./run_pipeline.pl -ListPlugins
```

```
./run_pipeline.pl -ListPlugins -usage true
```

Example Tassel Pipeline…

```
./run_pipeline.pl -importGuess maize.hmp.txt
-KinshipPlugin -method Centered_IBS -endPlugin
-export maize_kinship.txt -exportType SqrMatrix
```



filename:
maize.hmp.txt

method:
Centered_IBS

filename: maize_kinship.txt
format: SqrMatrix

FileLoad Plugin

Kinship Plugin

Export Plugin

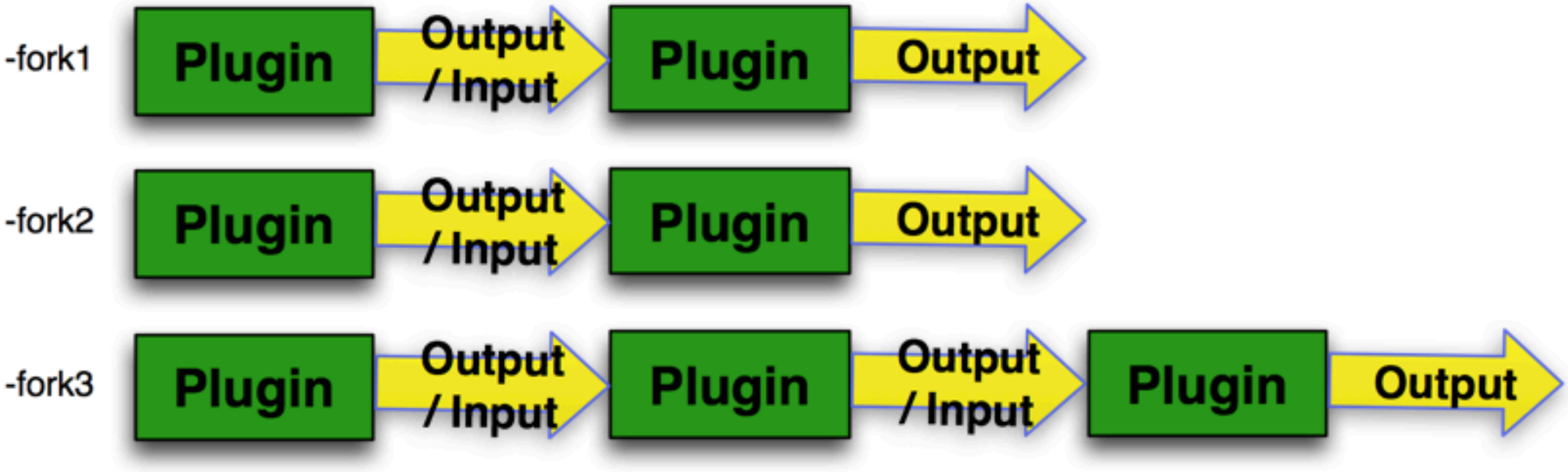
maize_kinship.txt

Tassel Pipeline Forks…

- Pipelines can consist of **multiple** Pipeline Forks (i.e. **Sub-Pipelines**).
- Defined by **-fork** or -combine flags. (Not needed if only one fork).
- **-combine** only used when combining output from multiple Plugins, as input to a single Plugin.
- Each Fork has a **name** (i.e. -fork**1**, -combine**A**).
- **Output** from a Sub-Pipeline can be used as **Input** to a another Sub-Pipeline by referencing with the flag -input. (i.e. -input**1**)
- Each Sub-Pipeline (i.e. -fork) runs in it’s **own CPU Process** (i.e. Thread)

Tassel Pipeline Forks…

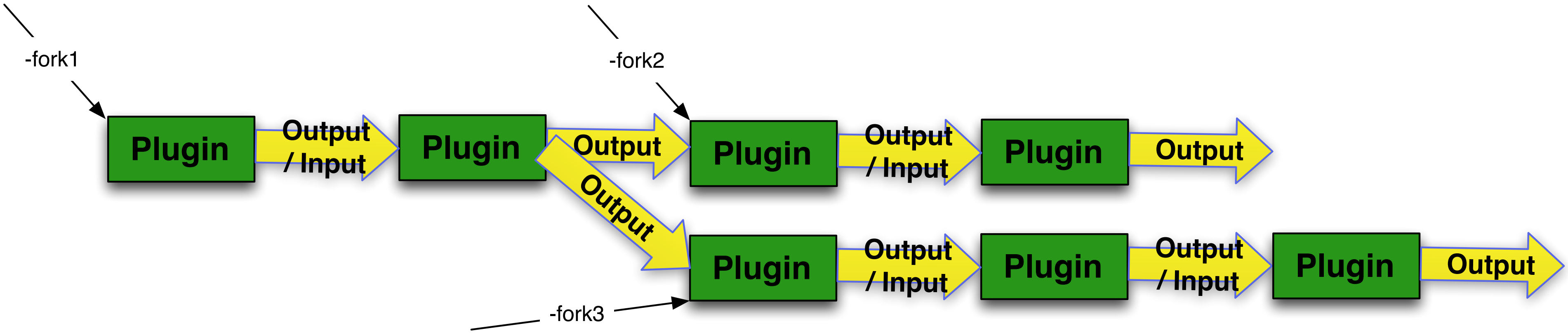
```
./run_pipeline.pl -fork1 <plugin> <plugin> -fork2
<plugin> <plugin> -fork3 <plugin> <plugin> <plugin>
```



Tassel Pipeline Inputs…

(Output Used as Input Twice)

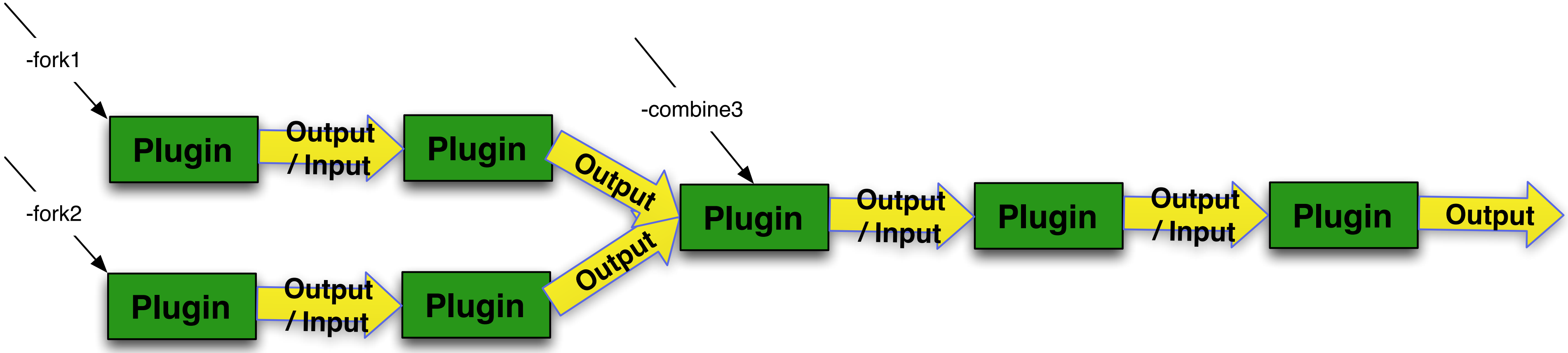
./run_pipeline.pl -fork1 <plugin> <plugin> -fork2 <plugin> -input1 <plugin> -fork3 <plugin> -input1 <plugin> <plugin>



Tassel Pipeline Inputs…

(Two Outputs Combined into One Input)

./run_pipeline.pl -fork1 <plugin> <plugin> -fork2 <plugin> <plugin> -combine3 -input1 -input2 <plugin> <plugin> <plugin>



Tassel Pipeline Inputs…

(Incorrect Usages!)

This -input1 doesn’t follow a Plugin or a -combine flag.

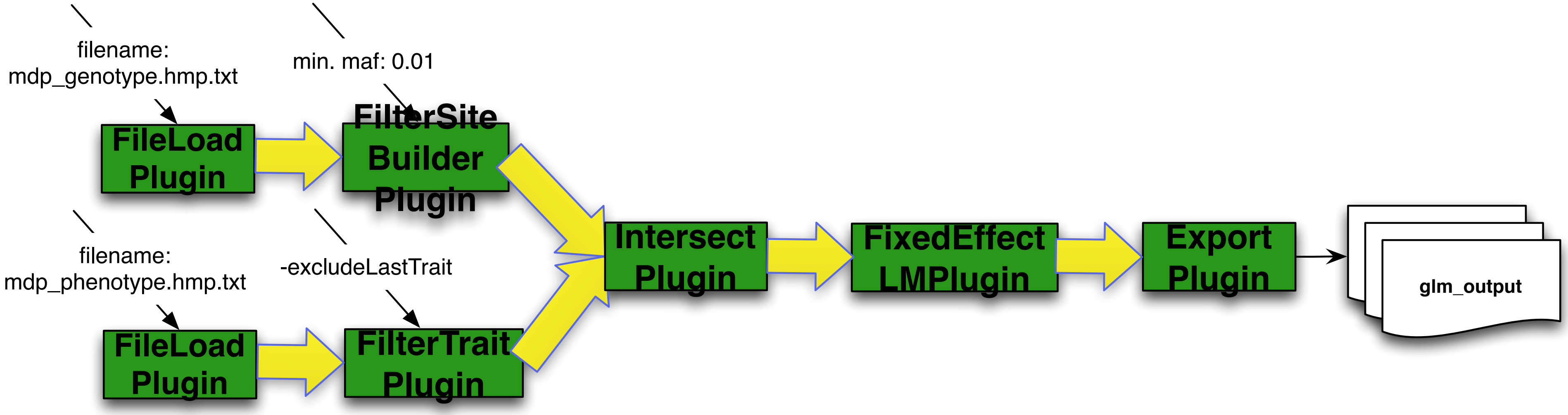
./run_pipeline.pl -fork1 <plugin> <plugin> -fork2
-input1 <plugin> <plugin>

pluginB getting input from pluginA and -input1

./run_pipeline.pl -fork1 <plugin> <plugin> -fork2
<pluginA> <pluginB> -input1

Tassel Pipeline GLM Example…

```
./run_pipeline.pl -fork1 -importGuess
mdp_genotype.hmp.txt -FilterSiteBuilderPlugin
-siteMinAlleleFreq 0.01 -endPlugin -fork2
-importGuess mdp_phenotype.txt -excludeLastTrait
-combine3 -input1 -input2 -intersect
-FixedEffectLMPlugin -endPlugin -export glm_output
```



Tassel Pipeline Options…

Run in debug mode

```
./run_pipeline.pl -debug [<filename>] …
```

Set maximum heap size

```
./run_pipeline.pl -Xmx10g …
```

Get usage for a plugin

```
./run_pipeline.pl <Plugin Name> -help
```