

# Quick Start Guide for FStitch

Michael Gohde

December 6, 2017

## Abstract

FStitch is a tool that learns and generates annotations for regions of transcription on GRO or ChIP-seq data. This document exists to provide a quick reference on how to invoke and properly use the new interface for FStitch.

## 1 Invoking FStitch

FStitch is designed to be run from a command line. All modes implemented by FStitch accept arguments in the following way:

`FStitch <command> [arguments]`

There are currently two commands usable by FStitch:

Command	Description
train	This command instructs FStitch to attempt to generate a set of trained weights based on a human-created file marking regions of transcription given a bedgraph.
segment	This command instructs FStitch to use a set of trained weights to generate predictions on regions of transcription given a bedgraph.

## 2 Training

The first step in using FStitch is to have it analyze a set of manually labeled regions of interest within a genome to determine a set of training weights. This allows FStitch to later predict regions of transcription

within a genome given a bedgraph file. These learned weights may be inferred from either a positive or negative strand's data, but not both. FStitch attempts to automatically infer which strand a given training set was created for, though this process is imperfect.

In order to perform training, it is necessary to pass, at minimum, the following list of parameters to FStitch:

1. *-t training.bed* FStitch needs a set of reference annotations so that it can learn how to mark regions given an input bedgraph.
2. *-r input.BedGraph* In order to determine what each label means in context, FStitch needs the input data from which reads were labeled.
3. *-o outputfilename* Once FStitch generates a set of weights, it needs to store them in an output file. This parameter allows you to specify that file.

It is further useful to manually specify which strand of data was used to create the training data provided. In order to do this, use the *--strand* parameter with one of the following options:

1. *+* Assume that training data covers the positive strand.
2. *-* Assume that training data covers the negative strand.
3. *both* While not yet implemented, this option will imply that the given training file has data for both the positive and negative strand. This will be accomplished through a BED6 file.

These parameters can be used to form the following command line:

```
FStitch train -t training.bed -r input.BedGraph -o outputfilename.out
```

The following is an example of a command line with an explicit strand specification:

```
FStitch train -t training.bed -r input.BedGraph -o outputfilename.out --strand -
```

Please note that the above parameters may be specified in any order as long as they follow the 'train' command. There are a number of additional parameters and modes that can be specified. To see these, either run FStitch without any arguments or consult the full users guide.

### 3 Segmentation

The second step in using FStitch is to have it predict regions of transcription given a weights file and a bedgraph. It is important to note that a given set of training weights can be applied to both positive and negative read data. By default, if presented with an input reads histogram containing both positive and negative strand data, FStitch will use the given training weights on both sets of data within that file. This requires the following set of parameters:

1. *-r input.BedGraph* It is necessary to have an input dataset to annotate.
2. *-w weights.out* This is the weights file generated in the previous step.
3. *-o output.bed* Once FStitch generates a set of annotations, it needs to store them in an output file readable by various genome viewers. This parameter allows you to specify that file.

These parameters can be used to form the following command line:

```
FStitch segment -r input.BedGraph -w weights.out -o output.bed
```

Please note that the above parameters may be specified in any order as long as they follow the ‘segment’ command.