

# Falco Tutorial

*Ho Lab*

*2019-11-29*



# Contents

<b>1</b>	<b>About the Workshop</b>	<b>5</b>
<b>2</b>	<b>Getting Started</b>	<b>7</b>
2.1	Prerequisites . . . . .	7
2.2	AWS setup . . . . .	7
<b>3</b>	<b>Reference Genome</b>	<b>11</b>
3.1	Sync the reference genome index from s3 bucket . . . . .	11
3.2	Check reference genome index from s3 bucket . . . . .	11
<b>4</b>	<b>Read Data</b>	<b>13</b>
4.1	Obtain the Read Data . . . . .	13
<b>5</b>	<b>Create AMI for Falco cluster nodes</b>	<b>15</b>
5.1	Have a look at the Custom AMI for Falco framework with tools pre-installed . . . . .	15
<b>6</b>	<b>Run the tutorial</b>	<b>17</b>
6.1	Give values for S3 bucket and User Name . . . . .	17
6.2	Launch the AWS EMR cluster . . . . .	18
6.3	Monitor the EMR Cluster . . . . .	18
6.4	Upload the Manifest file . . . . .	19
6.5	Launch the Split job . . . . .	19
6.6	Launch the Pre-processing job . . . . .	19
6.7	Launch the Analysis job . . . . .	20

6.8	Monitor Steps . . . . .	20
6.9	Monitor S3 files . . . . .	20
6.10	Download results and Terminate Cluster . . . . .	21

# Chapter 1

## About the Workshop

Falco is a software bundle that enables bioinformatic analysis of large-scale transcriptomic data by utilizing public cloud infrastructure. The framework currently provide supports for single cell RNA feature quantification, alignment and transcript assembly analyses.

This tutorial guides the user through the steps necessary to perform an analysis. It is assumed that any shell commands are executed using a bash shell.

Note that this tutorial will require you to either create a new AWS S3 bucket, or use an existing bucket. As a result, these instructions will require you to **replace occurrences of the word yourbucket (or [YOUR BUCKET]) with the name of the bucket you have created or chosen to use** - at various points in the instructions.

This tutorial is estimated to take up to 45 mins to complete. There is a step in this tutorial that launches the necessary computing resources on AWS. This step takes approximately 20 minutes to complete. Once verified that this step has started, the user can come back to the tutorial in 20 minutes to proceed with the remainder of the tutorial.

It is estimated that the cost of this tutorial, charged to your AWS account, will be less than \$5 USD. Make sure you follow the instructions to terminate the session once you have completed the tutorial.



## Chapter 2

# Getting Started

This is an example of how you may give instructions on setting up your project locally. To get a local copy up and running follow these simple example steps.

### 2.1 Prerequisites

It is assumed the user has downloaded the Falco framework and has extracted the files to a location of their choice onto a Linux operating system environment. This location will be referred to as the local resource, and the directory on the local resource that contains the LICENSE file from the Falco framework will be referred to as the home directory. Files related to this tutorial can be found in the tutorial directory.

Clone the repo

```
git clone https://github.com/VCCRI/Falco.git
```

Set path

```
export ..
```

### 2.2 AWS setup

#### 2.2.1 AWS account

If you are not part of an organisation that can provide you with IAM credentials, as described in Section 1.3 below, you will need to create an AWS account. Note that for this and subsequent AWS setup steps, this Getting Started with AWS documentation will be helpful.

### 2.2.2 Obtain an AWS secret key

If you are not the AWS administrator, ask your AWS administrator for AWS Access Key ID and Secret Access Key.

### 2.2.3 Install the AWS Linux Client (AWS CLI)

The AWS CLI is a command line tool that interfaces to the AWS resources such as EC2 instances, EMR clusters, and S3 storage buckets.

```
sudo pip install awscli
```

### 2.2.4 Configuring AWS Linux Client

Once installed, an initial configuration is required:

```
aws configure
```

When prompted, enter your Access Key ID and Secret Access Key, and default AWS region. Press **Enter** when prompted for output type if you are not sure about this element.

### 2.2.5 Obtain an AWS EC2 key

Obtain the AWS EC2 key name that will be used to control access to the instances that comprise the EMR cluster. A key will have an extension `.pem`. If you are familiar with AWS, you may already have an AWS EC2 key. Otherwise, you may create a key with these instructions.

### 2.2.6 Obtain access to or create an AWS S3 bucket

Create an S3 bucket for use with Falco.

```
aws s3api create-bucket --bucket [YOUR BUCKET] --region us-west-2
```

Replace `[YOUR BUCKET]` with your own bucket name. Also replace `us-west-2` if you would prefer to use a different region. Once created, use the AWS CLI high level S3 commands to work with your bucket - e.g. copy files to or from your bucket, list the contents of your bucket, etc.

Additionally, AWS provides a web interface to S3 (and other services).



### 2.2.7 Test access to the AWS management console

If you did not set up access to AWS yourself, you may have been provided a user name and password to access the AWS management console. You will also require your account ID. The console can be accessed at the address (replace [My\_AWS\_Account\_ID] with your account ID):

```
https://[My_AWS_Account_ID].signin.aws.amazon.com/console/]
```

### 2.2.8 Create Default AWS EMR Roles

The following link shows the necessary information on how to set up Roles for AWS EMR: [Create Default IAM Roles for Amazon EMR](#).

Note that this is a one-time only step for your organisation - check that it has not already been completed.

```
aws emr create-default-roles
```

### 2.2.9 AWS EMR Permissions

Make sure your user has the necessary permissions for both AWS EMR and AWS S3. For the sake of completing this tutorial, if you are not in the AWS Administrator group, you could configure or request that your user have the following IAM policies:

- AmazonS3FullAccess
- AmazonElasticMapReduceFullAccess

These policies can be configured from the AWS IAM Management console.



## Chapter 3

# Reference Genome

### 3.1 Sync the reference genome index from s3 bucket

In this tutorial, a human genome will be used. The file is ~800M in size.

In this step, the index for both STAR from our s3 bucket is synced to your s3 bucket.

At a bash prompt from the `genome_ref` directory, execute the following command - first change *[YOUR BUCKET]* to your own bucket name:

```
aws s3 sync s3://[YOUR BUCKET]/falco-tutorial/genomes/hg38/genome_ref --exclude "*.fa" --exclude
# change back to work directory
cd ..
```

### 3.2 Check reference genome index from s3 bucket

The `hg38_sparse_ref/` directory should look something like:

```
chrLength.txt
chrNameLength.txt
chrName.txt
chrStart.txt
exonGeTrInfo.tab
exonInfo.tab
geneInfo.tab
```

Genome  
genomeParameters.txt  
SA  
SAindex  
sjdbInfo.txt  
sjdbList.fromGTF.out.tab  
sjdbList.out.tab  
transcriptInfo.tab

## Chapter 4

# Read Data

### 4.1 Obtain the Read Data

For this tutorial, a script - `get_data.sh` is provided in the tutorial directory that will download a small number of relatively small FASTQ files (the total download size is approximately 380M, and a total of 10 individual files). The files are from the freely accessible Sequence Read Archive (SRA) database. To copy download and copy the files to a specified AWS S3 location, use the command (issued from your work directory):

```
tutorial/get_data.sh s3://[YOUR BUCKET]/falco-tutorial/data
```

Replace `[path-to-tutorial-directory]` with the actual path to the tutorial directory and `[YOUR BUCKET]` with the name of your bucket.



## Chapter 5

# Create AMI for Falco cluster nodes

### 5.1 Have a look at the Custom AMI for Falco framework with tools pre-installed

Falco has an alternative way to start the cluster without needing to run the software install script when the cluster is launched by using custom AMI (by specifying `custom_ami_id` option in `emr_cluster.config`). You will need to first create an AMI using the code in [https://github.com/VCCRI/Falco/tree/master/source/ami\\_creator](https://github.com/VCCRI/Falco/tree/master/source/ami_creator), by first launching the EC2 instance using `launch-ec2.sh` script (and then making sure that everything is installed properly), followed by running the `create-image.sh` script. When you run the `create-image.sh` script, it will create a new file called `custom_ami_id.txt` which will contain the custom AMI ID you will need to put into the `custom_ami_id` option in `emr_cluster.config` file. Once you have specified the `custom_ami_id`, you can set `bootstrap_scripts` option to just `copy_reference.sh` (instead of `install_software.sh`, `copy_reference.sh`).





## Chapter 6

# Run the tutorial

Change to Falco home directory.

```
vi tutorial/emr_cluster.config
```

Replace `ami_id` with the actual `ami_id`

### 6.1 Give values for S3 bucket and User Name

The tutorial files have a number of locations that have a placeholder called `username` and `yourbucket`. Change these placeholders to your values. In a bash shell, from the *Falco* home directory, edit and submit the following command:[]

```
# preplace the bracketed sections with your details
sed -i.bak 's/username/[YOUR USER NAME]/g ; s/yourbucket/[YOUR BUCKET NAME]/g' tutorial/*.config
# EXAMPLE ONLY: if your user name is "fred" and your AWS bucket is "falco-test", then you would use
#sed -i.bak 's/username/fred/g ; s/yourbucket/falco-test/g' tutorial/*.config
```

[] Additionally, if your AWS S3 region is not *us-west-2*, issue the following command:

```
sed -i.bak.reg 's/us-west-2/[YOUR REGION]/g' tutorial/*.config
# change [YOUR REGION] to the name of your region
# EXAMPLE (if your region is us-west-1): sed -i.bak.reg 's/us-west-2/us-west-1/g' tutorial/*.config
```

The original `.config` files in the tutorial directory will now have the file extension `.config.bak` should you wish to restore the original files.

## 6.2 Launch the AWS EMR cluster

[ ] First open the `filetutorial/emr_cluster.config` and check that the `[EMR_nodes]` section is similar to the following:

```
[EMR_nodes]
key_name = yourkey
service_role = EMR_DefaultRole
instance_profile = EMR_EC2_DefaultRole
master_instance_type = r3.4xlarge
master_instance_count = 1
core_instance_type = r3.4xlarge
core_instance_count = 2
core_instance_spot = True
core_instance_bid_price = 1
```

The `[EMR_nodes]` section contains a line that starts with `key_name =`. The computing resources created by the AWS EMR framework uses public-key cryptography to encrypt and decrypt login information. You need to supply your key name here. For example, if your encryption key file is `my-key-name.pem`, the corresponding line in the configuration file should read `key_name = my-key-name`. [ ] Go ahead and edit this entry - enter your key name. [ ] Also be aware of the charges that you may incur from AWS for the creation of this cluster. The master instance will be an on-demand type, whilst the two core instances are spot instance types. It is estimated the cost of the cluster for this tutorial, if terminated within 1 hour, will be less than \$5 USD. This cost is based on the AWS region `us-west-2` - US West (Oregon).

When ready, in the home directory of the *Falco* code, issue the following command to start the cluster: [ ]

```
python3 launch_cluster.py --config tutorial/emr_cluster.config
```

When the command is processed, the user will receive a response, of the form:

```
Cluster has been launched with ID j-1FDPU9CHN79W9
```

Make a note of your cluster ID for future reference.

## 6.3 Monitor the EMR Cluster

[ ] Monitor the status of the EMR cluster via the AWS EMR console. When the status of your cluster is Ready, you may proceed with the steps required for completing the analysis.

Click on your cluster to obtain more information about your cluster.

[ ] Since the nodes that are launched as part of this cluster use AWS spot instance types, it is possible that the market price for the instances exceeds the bid price. If this is the case, then the cluster will not start until the market price falls below the bid price. You can monitor the market price for the spot instances via the AWS EC2 Management Console. You can decide if you want to terminate your EMR cluster and either try again later, or modify the bid price in the file `tutorial/emr_cluster.config`.

## 6.4 Upload the Manifest file

Falco requires a manifest file to list the FASTQ filenames representing the data input. The required format is a tab delimited text file. This file has been provided for this tutorial and is located at `tutorial/data.manifest`. Upload the manifest file to your AWS S3 bucket: [ ]

```
# issue this command from the Falco home directory
aws s3 cp tutorial/data.manifest s3://[YOUR BUCKET]/falco-tutorial/data.manifest
# replace [YOUR BUCKET] with the name of your bucket
```

[ ]The following three instructions that launch jobs can be issued one after the other - without waiting for the previous job to finish. The EMR framework will launch jobs in order, and only after the previous job has completed.

## 6.5 Launch the Split job

The split job takes the original data and splits it into smaller sized files for more efficient processing by Falco. The original input data stored on AWS S3 will not be removed. The modified data will be stored in a new AWS S3 location as specified in the configuration file `tutorial/split_job.config`. Type the following command at a command prompt in the Falco home directory:[ ]

```
python3 submit_split_job.py --config tutorial/split_job.config
```

## 6.6 Launch the Pre-processing job

In Falco, the pre-processing step is optional. However, for this tutorial, an example pre-processing script is provided, and this step is compulsory to complete the tutorial as configured.

First examine the configuration file `tutorial/preprocessing_job.config`. The config file specifies which bash scripts are used for the pre-processing. You may wish to also examine these scripts to see how the pre-processing works in this case. Type the following at a command prompt from the Falco home directory: []

```
python3 submit_preprocessing_job.py --config tutorial/preprocessing_job.config
```

## 6.7 Launch the Analysis job

This is the main analysis job that processes the pre-processed data to determine the counts of features. The output be two .csv files: the actual counts of features, and a separate file detailing quality assurance statistics relating to these counts.

The configuration file `tutorial/analysis_job.config` contains the settings for the analysis job with STAR as the aligner and featureCount for quantification - including any extra parameters for the tools used. The configuration file also specifies the AWS S3 location for the final output .csv files. Enter the following at a command prompt from the Falco home directory: []

```
python3 submit_analysis_job.py --config tutorial/analysis_job.config
```

[] To change the alignment and/or quantification tool used in the analysis job, simply modify the `aligner_tool` and `counter_tool` option in the analysis configuration file (`analysis_job.config`).

## 6.8 Monitor Steps

Note that, as long as the above three steps (split, preprocess, and analysis) are entered in that order, the steps may be entered one after the other - without having to wait until the previous step finishes. The EMR framework ensures that a step does not start until the previously queued step has completed.

[] Monitor the status of each step using the AWS EMR console - in the Step section.

## 6.9 Monitor S3 files

Use the AWS S3 Console to monitor files in your S3 bucket.

## 6.10 Download results and Terminate Cluster

To download a file from AWS S3 to your current directory, edit the following command with your details and execute at a shell command line: []

```
aws s3 cp s3://[YOUR BUCKET]/falco-tutorial/[YOUR USER NAME]/analysis/samples_expression.csv .  
# you can also list all the AWS S3 files that begin with a particular prefix:  
#aws s3 ls s3://[YOUR BUCKET]/falco-tutorial/[YOUR USER NAME]/analysis/
```

Alternatively, you could see a listing of the results via the AWS S3 Console by navigating to your bucket and output location.

[]As AWS charges by the hour for usage of its services, the user should terminate their cluster when finished. This is done by selecting your cluster on the AWS EMR console *Cluster List*, and clicking the *Terminate* button.