

Anglican Getting Started Guide

PPAML Summer School 2016

1 Introduction

This guide and associated set of programming exercises will get you up and programming using the Anglican programming language and system <http://www.robots.ox.ac.uk/~fwood/anglican> in a very short period of time.

If you go through all the materials you will end up comfortable programming in Clojure and Anglican, and, more importantly, have an idea about with how to program in probabilistic programming languages in general.

2 Installing Anglican

Anglican is a probabilistic programming language that compiles to Clojure which subsequently compiles to JVM bytecode. For this reason you need the Java and Clojure ecosystems installed on either your own personal computer or on a machine into which you can ssh, and, in the latter case, to which you can open socket (http) connections.

2.1 Java Prerequisites

Clojure depends on having a recent Java Development Kit installed. Windows and Mac OS X users can download Java DK installers from <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> (*version 8u73 is fine*). Linux users who do not already have Java installed can install from their package managers:

```
# Debian/Ubuntu
sudo apt-get install default-jdk
```

```
# Fedora
sudo yum install java-1.7.0-openjdk
```

2.2 Install Leiningen

Leiningen is a self-installing automated Clojure project management system. You must install Leiningen from <http://leiningen.org/>. “lein” (short for

Leiningen) is a self installing script as well as the primary means of invoking both Anglican and Clojure read eval print loops (REPL). Fortunately “lein” is trivial to install in *nix environments (see below). **Note that Leiningen version >2.x is required; the version in GNU-Linux package repositories may be quite a bit out of date.**

The following sequence of commands will, by-in-large, install and make “lein” runnable on your system. For Unix experts the particulars are obvious and simply involve downloading and running a shell script.

```
# Download lein to ~/bin
mkdir ~/bin
cd ~/bin
wget https://raw.githubusercontent.com/technomancy/leiningen/stable/bin/lein

# Make executable
chmod a+x ~/bin/lein

# Add ~/bin to path
# Note: Mac OS X users should replace ".bashrc" with ".profile"
echo 'export PATH="$HOME/bin:$PATH"' >> ~/.bashrc

source ~/.bashrc

# Run lein
lein
```

Windows users have it just as easy. They just can use the Leiningen installer, which installs the latest version of Leiningen: <http://leiningen-win-installer.djpowell.net/>.

2.3 Download Anglican exercises

The exercises themselves can be cloned from a BitBucket repo (requires a BitBucket accounts):

<https://bitbucket.org/probprog/ppaml-summer-school-2016>

When you have managed to do all this successfully then, in effect, you will have a Leiningen (Clojure) project sitting locally on your machine. Within it you should try to start a web-based, Anglican-enabled Gorilla REPL: ¹

```
# replace "ppaml-summer-school-2016" with name of the unzipped directory
# containing the exercises
cd ppaml-summer-school-2016/exercises
lein gorilla :port 8990
```

which will start a web service on port 8990 which allows you to view, edit, and run the different Anglican example programs.

If it says “Could not reserve enough space...,” no worries. Just reduce the amount of the memory for the heap by modifying the appropriate line in `/project.clj` to something like this:

```
:jvm-opts ["-Xmx1g" "-Xms1g"]
```

Use a web browser (Chrome, e.g.) to open `http://localhost:8990/worksheet.html`. Using the menu on the top right (Fig. 1) you should be able to open and interactively run the exercises, using the “Load a worksheet” command.



Figure 1: Click here to load worksheets.

Start by loading

```
exercises/worksheets/intro-to-clojure/01-clojure-overview.clj.
```

On a Mac you can step through (and execute) the cells in the worksheet by pressing `shift+enter` (also accessible from the menu; other OS key combos may be different). This worksheet introduces you to functional programming and Clojure.

The following joke indicates where you should stop on Monday afternoon, after ensuring that your machine is configured and ready to go for the next day.

Q: Why do programmers always mix up Halloween and Christmas?

A: Because Oct 31 == Dec 25!

¹Users installing on a server will instead run `lein gorilla :ip 0.0.0.0`.

The worksheet you ran before stopping at the preceding joke contains all the Clojure background required to complete the exercises in the following worksheets, which will be introduced and worked through over the week in the following order:

1. Tuesday 10am-12pm

- (a) `/worksheets/intro-to-clojure/01-clojure-overview.clj`
- (b) `/worksheets/intro-to-clojure/02-clojure-exercises.clj`

2. Tuesday 3:30pm-5pm

- (a) `/worksheets/intro-to-inference/01-introduction-to-inference.clj`
- (b) `/worksheets/intro-to-inference/b1-arithmetic-function-induction.clj`

3. Wednesday 10am-12pm; 2:30pm-5pm

- (a) `/worksheets/intro-to-anglican/01-hello-world.clj`
- (b) `/worksheets/intro-to-anglican/02-gaussian.clj`
- (c) `/worksheets/intro-to-anglican/03-physics.clj`
- (d) `/worksheets/intro-to-anglican/04-poisson-trace.clj`
- (e) `/worksheets/intro-to-anglican/b1-coordination.clj`

3 Optional

3.1 Clojure Programming

Ideally you would be a Clojure programmer, or, at least, a functional programming expert in advance of learning Anglican. It is not necessary however, as these materials are self contained. Familiarising yourself with Clojure even beyond the material provided may help your overall experience however.

In general, the Clojure main website <http://clojure.org/> has links to a large number of language learning resources, in particular <http://clojure-doc.org/articles/tutorials/introduction.html>.