Run EQTL with batchtools SSH cluster functions

Credentials

By default, boto3 loads credentials from `~/.aws/credentials`. To use different credentials, set the environment variable:

```
export AWS_SHARED_CREDENTIALS_FILE=/Users/spollack/.aws/credentials.new-group-acct
```

AMI

Public AMI ami-e3c0bf99 is Ubuntu 16.04, with R-3.4.3, Bioconductor 3.6, batchtools and the EQTL dependencies installed. It was made with these commands:

```
sudo apt-get update
sudo apt-get install -y r-base-core                    # brings in the R dependencies
sudo apt-get install -y libcurl4-openssl-dev           # devtools dependency
sudo apt-get install -y libssl-dev                     # devtools dependency
sudo apt-get install -y nfs-kernel-server              # NFS server (also EFS dependency)
sudo apt-get install -y libxml2-dev                    # XML dependency
sudo apt-get install -y default-jdk                    # R-3.4.3 dependency
sudo apt-get install -y libmariadb-client-lgpl-dev     # RMySQL dependency

# Install R-3.4.3 from source
wget https://cran.r-project.org/src/base/R-3/R-3.4.3.tar.gz
gunzip R-3.4.3.tar.gz
tar xvf R-3.4.3.tar
cd R-3.4.3
sudo ./configure --prefix=/usr/local/lib/R-3.4.3 --with-x=no
sudo make all
sudo make install

# Edit /etc/environment to put R-3.4.3 in the path:
PATH="/usr/local/lib/R-
3.4.3/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local
/games"
# Comment out env_reset and secure_path (sudo visudo) so sudo also finds R-3.4.3

sudo /usr/local/lib/R-3.4.3/bin/R      # install to system library
> install.packages("devtools", repos="http://lib.stat.cmu.edu/R/CRAN")
> install.packages("XML", repos="http://lib.stat.cmu.edu/R/CRAN")       # rtracklayer
dependency
```

```
> source("https://bioconductor.org/biocLite.R")
> biocLite();
> biocLite("gQTLstats");   # ldblock and VariantAnnotation installed with this
> biocLite("geuvPack")

# use sampoll/batchtools to avoid the SSH linux-helper path quoting problem.
> library(devtools)
> install_github("sampoll/batchtools")
```

Note: The quoting in Worker.R at line 49 does not work on this system. (It is an open question whether it works on any other system.) To make this work, I use the path to the `linux-helper` script in the local build, on the assumption that all of the R source trees are the same, which is true as long as we are using the same AMI.

This means we need to install and use the batchtools from my GitHub repository.

Configuration

Bioconductor is too large for a t2.micro, so we use a t2.small. A sample config file looks like this. The config file name is the first command line argument.

```
[clusterdef]
name = cluster-eqtl
vpc = vpc-576d222f
nnode = 2
key = sam-new-account
type = t2.small
ami = ami-ec30bf99
```

The cluster has to be set up in two parts, because the nodes are not ready to accept connections on the ssh port until some time after boto3 classifies them as "running." The second command line argument is "1" for the first part of the setup (launch nodes) and "2" for the rest (once the nodes are ready for ssh.)

Setting nnode to 2 will result in one head node (which does not do work) and one compute node.

(A possible solution: https://stackoverflow.com/questions/7405598/process-for-telling-when-a-new-ec2-host-can-be-connected-to)

This cluster will use NFS. (Configuration file with EFS is below.) When the cluster is set up, `run.py` prints out the IP Addresses. For example:

```
$ python run.py eqtl.nfs.config 1
# wait until console shows node status checks are done
$ python run.py eqtl.nfs.config 2
# program prints output, ending in
Head node: public IP Address: 54.159.134.146  Private IP Address: 172.31.92.105
Compute node: public IP Address: 54.91.232.33  Private IP Address: 172.31.80.128
```

Run EQTL

ssh to the head node and cd to `/scratch`. The `batchtools.conf.R` file should already be there. sftp up `eqtl.R`:

```
# eqtl.R
library(gQTLstats)
library(ldblock)
library(VariantAnnotation)
library(geuvPack)
library(batchtools)

data(geuFPKM)
ss <- stack1kg()
v17 <- ss@files[[17]]
someGenes <- c("ORMDL3","GSDMB", "IKZF3", "MED24", "CSF3", "ERBB2",
               "GRB7", "MIEN1", "GSDMA", "THRA", "MSL1")
se17 <- geuFPKM[ which(rowData(geuFPKM)$gene_name %in% someGenes),]

n <- 10
vr0 <- 39.5e6
vr1 <- 40.5e6
v <- seq(vr0, vr1, length=n+1)
vl <- zipup(v[1:n],v[2:(n+1)]-1)

run.job <- function(v, se, vcf)  {
  library(gQTLstats)
  library(ldblock)
  library(VariantAnnotation)
  library(geuvPack)

  vr <- GRanges("17", IRanges(start=v[1], end=v[2], names=c("range")))
  results <- AllAssoc(se, vcf, vr)
}

concat.job <- function(gr1, gr2)  {
  gr <- c(gr1,gr2)
}
```

Run from R command prompt:

```
R
> source("eqtl.R")
> system("rm -Rf ./registry")    # in case there is an old registry
> reg <- makeRegistry();
> batchMap(fun = run.job, as.list(vl), more.args = list(se=se17, vcf=v17))
> submitJobs()
> waitForJobs()
> all.results <- reduceResults(fun = concat.job)
> save(all.results, file="all.results.RData")
```

Run with EFS

The EFS is `fs-e9a39da0`, in `VPC-5763222f`. DNS name: `fs-e9a39da0.efs.us-east-1.amazonaws.com`

There are two security groups for the EFS; one for the mount targets and one for the nodes that will mount it. The node SG is called `bioc-efs-node-sg` and the mount SG is called `bioc-efs-mount-sg`. on port 22 from anywhere. `bioc-efs-mount-sg` allows incoming NFS traffic on port 2049 from `bioc-cfn-sg`.

In the EFS there is a top-level directory called /btrun which has group and user owner `ubuntu:ubuntu` and read-write-execute permissions set to 775. This will allow any node based on Ubuntu to read from and write to a batchtools registry.

(`bioc-efs-node-sg` allows incoming traffic on any port from `bioc-efs-mount-sg` and)

Config File

```
[clusterdef]
name = cluster-eqtl
vpc = vpc-576d222f
nnode = 3
key = sam-new-account
type = t2.small
ami = ami-e3c0bf99
sgn = bioc-efs-node-sg
efs = fs-e9a39da0.efs.us-east-1.amazonaws.com
```

Run EQTL

ssh to the head node and cd to `/efs/btrun`. `eqtl.R` is already there. Execute the same commands at the R command prompt as in the NFS cluster.

Note:

When submitting jobs, SSH asks for confirmation to continue connecting once per host. This is annoying. Something should go into the known_hosts file to prevent this.