Introduction to *BatchtoolsParam*

Nitesh Turaga[1]***, Martin Morgan***[2]

[1]Nitesh.Turaga@RoswellPark.org

[2]Martin.Morgan@RoswellPark.org

Edited: March 22, 2018; Compiled: May 5, 2018

Contents

1 Introduction

The `BatchtoolsParam` class is an interface to the *batchtools* package from within *BiocParallel*. This aims to replace BatchjobsParam as *BiocParallel*'s class for computing on a high performance cluster such as SGE, TORQUE, LSF, SLURM, OpenLava.

2 Quick start

This example demonstrates the easiest way to launch a 100000 jobs using batchtools. The first step involves creating a `BatchtoolsParam` class. You can compute using 'bplapply' and then the result is stored.

```
library(BiocParallel)

## Pi approximation
piApprox <- function(n) {
    nums <- matrix(runif(2 * n), ncol = 2)
    d <- sqrt(nums[, 1]^2 + nums[, 2]^2)
    4 * mean(d <= 1)
}

piApprox(1000)

## [1] 3.16
```

```
## Apply piApprox over
param <- BatchtoolsParam()
result <- bplapply(rep(10e5, 10), piApprox, BPPARAM=param)

## Adding 10 jobs ...

## Submitting 10 jobs in 4 chunks using cluster functions 'Multicore' ...
```

```
## cleaning registry...

mean(unlist(result))

## [1] 3.142066
```

3 *BatchtoolsParam* interface

The `BatchtoolsParam` interface allows you to replace the BatchJobsParam interface, and allow more intuitive usage of your high performance cluster with *BiocParallel*.

The `BatchtoolsParam` class allows the user to specify many arguments to customize their jobs. Applicable to clusters with formal schedulers.

- workersThe number of workers used by the job.
- cluster We currently support, SGE, SLURM, LSF, TORQUE and OpenLava. The 'cluster' argument is supported only if the R environment knows how to find the job scheduler. Each cluster type uses a template to pass the job to the scheduler. If the template is not given we use the default templates as given in the 'batchtools' package. The cluster can be accessed by 'bpbackend(param)'.
- registryargs The 'registryargs' argument takes a list of arguments to create a new job registry for you `BatchtoolsParam`. The job registry is a data.table which stores all the required information to process your jobs. The arguments we support for registryargs are,

  'file.dir': Path where all files of the registry are saved. Note that some templates do not handle relative paths well. If nothing is given, a temporary directory will be used in your current working directory.

  'work.dir': Working directory for R process for running jobs.

  'packages': Packages that will be loaded on each node.

  'namespaces': Namespaces that will be loaded on each node.

  'source': Files that are sourced before executing a job.

  'load': Files that are loaded before executing a job.

```
registryargs <- batchtoolsRegistryargs(
    file.dir = "mytempreg",
    work.dir = getwd(),
    packages = character(0L),
    namespaces = character(0L),
    source = character(0L),
```

```
        load = character(0L)
    )
    param <- BatchtoolsParam(registryargs = registryargs)
    param

    ## class: BatchtoolsParam
    ##   bpisup: FALSE; bpnworkers: 4; bptasks: 0; bpjobname: BPJOB
    ##   bplog: FALSE; bpthreshold: INFO; bpstopOnError: TRUE
    ##   bptimeout: 2592000; bpprogressbar: FALSE
    ##   cluster type: multicore
    ##   template: NA
    ##   bpRNGseed: NA
    ##   bplogdir: NA
    ##   registryargs:
    ##     file.dir: mytempreg
    ##     work.dir: /tmp/RtmpiSDBQL/Rbuild24e5786148f6/BiocParallel/vignettes
    ##     packages: character(0)
    ##     namespaces: character(0)
    ##     source: character(0)
    ##     load: character(0)
    ##     make.default: FALSE
```

- template

  The template argument is unique to the `BatchtoolsParam` class. It is required by the job scheduler. It defines how the jobs are submitted to the job scheduler. If the template is not given and the cluster is chosen, a default template is selected from the batchtools package. We recommend that the user define a template which works on their cluster, and supply a path to the template argument.

- log

  The log option is logical, TRUE/FALSE. If it is set to TRUE, then the logs which are in the registry are copied to directory given by the user using the `logdir` argument.

- logdir

  Path to the logs. It is given only if `log=TRUE`.

- resultdir

  Path to the directory is given when the job has files to be saved in a directory.

4 Use cases

As an example for a BatchtoolParam job being run on an SGE cluster, we use the same `pi Approx` function as defined earlier. The example runs the function on 5 workers and submits 100 jobs to the SGE cluster.

Example of SGE with minimal code:

```
library(BiocParallel)

## Pi approximation
```

```
piApprox <- function(n) {
    nums <- matrix(runif(2 * n), ncol = 2)
    d <- sqrt(nums[, 1]^2 + nums[, 2]^2)
    4 * mean(d <= 1)
}

template <- system.file(
    package = "BiocParallel",
    "unitTests", "test_script", "test-sge-template.tmpl"
)
param <- BatchtoolsParam(workers=5, cluster="sge", template=template)

## Run parallel job
result <- bplapply(rep(10e5, 100), piApprox, BPPARAM=param)
```

Example of SGE demonstrating some of `BatchtoolsParam` methods.

```
library(BiocParallel)

## Pi approximation
piApprox <- function(n) {
    nums <- matrix(runif(2 * n), ncol = 2)
    d <- sqrt(nums[, 1]^2 + nums[, 2]^2)
    4 * mean(d <= 1)
}

template <- system.file(
    package = "BiocParallel",
    "unitTests", "test_script", "test-sge-template.tmpl"
)
param <- BatchtoolsParam(workers=5, cluster="sge", template=template)

## start param
bpstart(param)

## Display param
param

## To show the registered backend
bpbackend(param)

## Register the param
register(param)

## Check the registered param
registered()

## Run parallel job
result <- bplapply(rep(10e5, 100), piApprox)

bpstop(param)
```

5 sessionInfo()

```
toLatex(sessionInfo())
```

- R version 3.5.0 (2018-04-23), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Running under: Ubuntu 16.04.4 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.7-bioc/R/lib/libRblas.so
- LAPACK: /home/biocbuild/bbs-3.7-bioc/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: BiocParallel 1.14.1
- Loaded via a namespace (and not attached): BiocStyle 2.8.0, R6 2.2.2, Rcpp 0.12.16, assertthat 0.2.0, backports 1.1.2, base64url 1.3, batchtools 0.9.8, brew 1.0-6, checkmate 1.8.5, compiler 3.5.0, crayon 1.3.4, data.table 1.11.0, debugme 1.1.0, digest 0.6.15, evaluate 0.10.1, highr 0.6, htmltools 0.3.6, knitr 1.20, magrittr 1.5, parallel 3.5.0, prettyunits 1.0.2, progress 1.1.2, rappdirs 0.3.1, rmarkdown 1.9, rprojroot 1.3-2, stringi 1.2.2, stringr 1.3.0, tools 3.5.0, withr 2.1.2, yaml 2.1.19