VariantBench – benchmarking variant access and analysis

Vincent J. Carey, stvjc at channing.harvard.edu

February 03, 2017

Contents

1 Introduction

We will illustrate a simple use of the harness for multiple approaches to VCF access. We have a local version of the Tabix-indexed VCF for chr17 for 1000 genomes. We'll set up packages and path.

```
library(VariantBench)
library(GenomicRanges)
loc17 = "/Users/stvjc/Research/VCF/ALL.chr17.phase3_shapeit2_mvncall_integrated_v5.20130502.genotypes.vcf.
```

An illustrative *closure* is provided that encapsulates the information about the VCF to be processed.

```
useScanVcfClo
## function (vcffile)
## function(gr, times) {
##     parm = ScanVcfParam(which = gr, fixed = NA, info = NA, geno = "GT")
##     timing = microbenchmark(dat <- scanVcf(vcffile, param = parm),
##         times = times)
##     list(timing = timing, request = gr, obj.size = object.size(dat))
## }
## <environment: namespace:VariantBench>
```

We bind the file path to the VCF processing function.

```
useScanVcf_local = useScanVcfClo(loc17)
useScanVcf_local
## function (gr, times)
## {
##     parm = ScanVcfParam(which = gr, fixed = NA, info = NA, geno = "GT")
##     timing = microbenchmark(dat <- scanVcf(vcffile, param = parm),
##         times = times)
##     list(timing = timing, request = gr, obj.size = object.size(dat))
## }
## <environment: 0x7fdd340b1698>
ls(environment(useScanVcf_local))
## [1] "vcffile"
```

Now run the harnessed processing function.

```
vbHarness(GRanges("17", IRanges(16e6, 16.01e6)),
    list(useScanVcf=useScanVcf_local))
## $useScanVcf
## $useScanVcf$timing
## Unit: milliseconds
```

```
##                                   expr      min       lq     mean   median
##  dat <- scanVcf(vcffile, param = parm) 532.4994 539.9321 553.9965 552.8688
##        uq      max neval
##  558.6217 586.0608     5
##
## $useScanVcf$request
## GRanges object with 1 range and 0 metadata columns:
##       seqnames              ranges strand
##          <Rle>           <IRanges>  <Rle>
##   [1]       17 [16000000, 16010000]      *
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
##
## $useScanVcf$obj.size
## 5787352 bytes
```

We can collect information on multiple request types by iterating over ranges of various widths.

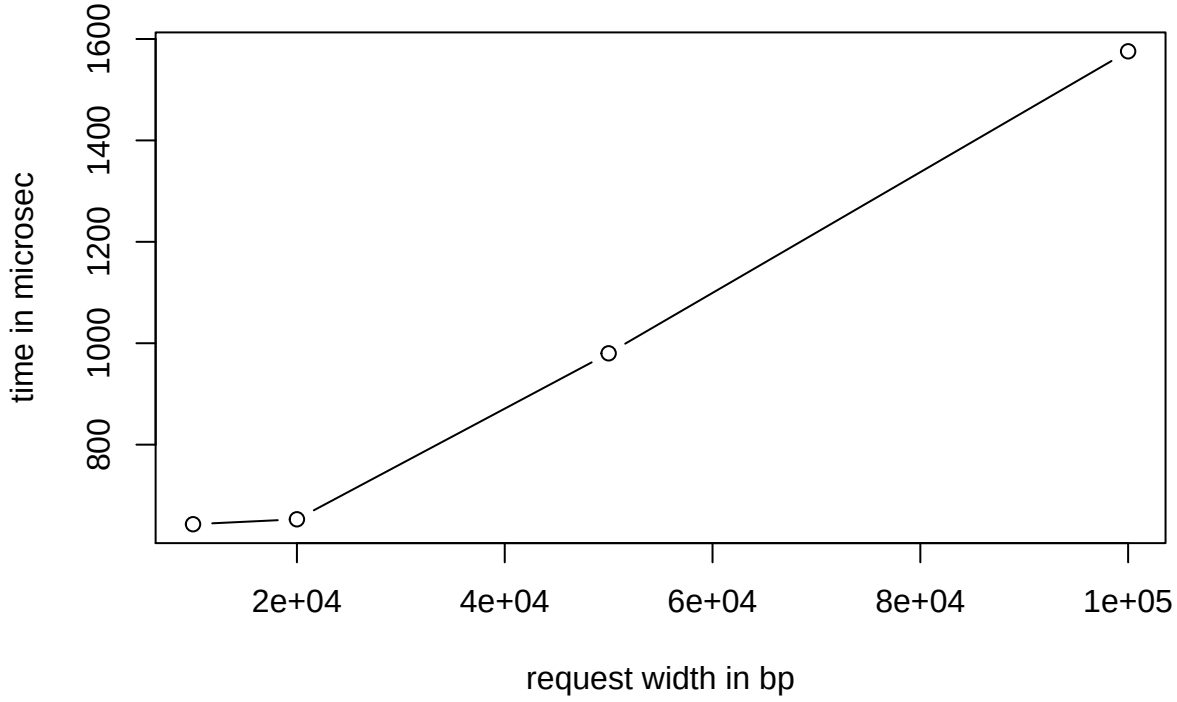
```
rngs = GRanges("17", IRanges(16e6, width=c(1e4, 2e4, 5e4, 1e5)))
multperf = lapply(rngs, function(r)
  vbHarness(r, list(useScanVcf=useScanVcf_local)))
```

The following overcomplicated functions extract key information about performance. These are fragile to details of the output of the method passed to the harness.

```
widths = function(x) sapply(x, sapply,
  function(y) width(y$request))
meantimes = function(y)apply(sapply(y,
    function(x) (x$useScanVcf$timing$time)), 2, mean)
```

This permits a plot like

```
plot( widths(multperf), meantimes(multperf)/10^6, type="b",
  xlab="request width in bp", ylab="time in microsec")
```


time in microsec
800
1000
1200
1400
1600
2e+04
4e+04
6e+04
8e+04
1e+05
request width in bp