

```

1 op = options()
2 options(stringsAsFactors = FALSE)
3 suppressMessages(library(RCurl))
4 suppressMessages(library(stringr))
5 suppressMessages(library(rvest))
6
7 # Get All Hisat2 ftp reffa url
8 get.hisat2_reffa.versions <- function() {
9   urls <- "ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/data/"
10  versions_final <- NULL
11  for (url in urls) {
12    web <- read_html(url, encoding = "UTF-8")
13    files.txt <- web %>% html_nodes("p") %>% html_text()
14    files.row <- str_split(files.txt, "\n") %>% .[[1]] %>% as.character()
15    files.row <- str_split(files.row, " ")
16    files <- lapply(files.row, function(x) return(x[str_detect(x,
"tar.gz"))])
17    versions <- str_extract(files, ".*tar.gz")
18    versions <- versions[!is.na(versions)]
19    versions <- str_extract(versions, ".*tar")
20    versions <- str_replace(versions, ".tar$", "")
21    versions_final <- c(versions_final, versions)
22  }
23  return(versions_final)
24 }
25
26 # Function get GMAP-GSNAP all versions
27 get.gmap.versions <- function() {
28   urls <- c("http://research-pub.gene.com/gmap/src/")
29   versions_final <- NULL
30   for (url in urls) {
31     web <- read_html(url, encoding = "UTF-8")
32     files.table <- web %>% html_nodes("table") %>% .[[1]] %>% html_table()
33     files <- files.table$Name
34     versions <- str_extract(files, "gmap-.*tar.gz")
35     versions <- versions[!is.na(versions)]
36     versions <- str_extract(versions, "gmap-.*tar")
37     versions <- str_replace(versions, ".tar$", "")
38     versions_final <- c(versions_final, versions)
39   }
40   return(versions_final)
41 }
42
43 # Function get Edena all versions
44 get.edena.versions <- function() {
45   urls <- c("http://www.genomic.ch/edena.php")
46   versions_final <- NULL
47   for (url in urls) {
48     web <- read_html(url)
49     hrefs <- web %>% html_nodes(".txt a") %>% html_attr(name = "href")
50     hrefs <- hrefs[!is.na(hrefs)]
51     hrefs <- hrefs[str_detect(hrefs, ".tar.gz|.zip")]
52     versions <- basename(hrefs)
53     versions <- str_replace_all(versions, ".tar.gz|.zip", "")
54     versions_final <- c(versions_final, versions)
55   }
56   return(versions_final)
57 }
58
59 # Function get fastqc all versions

```

```

60 get.fastqc.versions <- function() {
61   urls <- c("http://www.bioinformatics.babraham.ac.uk/projects/fastqc/")
62   versions_final <- NULL
63   for (url in urls) {
64     web <- read_html(url)
65     versions <- web %>% html_nodes("div ul li") %>% html_text()
66     versions <- versions[str_detect(versions, "Version")]
67     versions <- versions[str_detect(versions, "released")]
68     versions <- str_extract(versions, "Version [0-9][.0-9]*")
69     versions <- str_replace_all(versions, "Version ", "")
70     versions_final <- c(versions_final, versions)
71   }
72   return(versions_final)
73 }
74
75 # Function get Novoalign all versions
76 get.novoalign.versions <- function() {
77   urls <- "http://www.novocraft.com/support/download/"
78   versions_final <- NULL
79   for (url in urls) {
80     myheader <- c(`User-Agent` = "Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_0_1
like Mac OS X; ja-jp) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.0.5
Mobile/8A306 Safari/6531.22.7",
81     Accept =
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
82     `Accept-Language` = "en-us", Connection = "keep-alive", `Accept-
Charset` = "GB2312,utf-8;q=0.7,*;q=0.7")
83
84     web <- getURL(url = "http://www.novocraft.com/support/download/",
httpheader = myheader,
85     verbose = F)
86     web <- str_split(web, "\n") %>% .[[1]]
87     web <- web[str_detect(web, "download.php")]
88     web <- web[str_detect(web, "tar.gz")]
89     versions <- str_extract(web, "[0-9][.0-9]*")
90     versions_final <- c(versions_final, versions)
91   }
92   versions_final <- paste0("V", versions, c(".Linux3.0", ".Linux2.6",
".MacOSX"))
93   return(versions_final)
94 }
95
96 # Function get ssaha2 all versions
97 get.ssaha2.versions <- function() {
98   urls <- c("ftp://ftp.sanger.ac.uk/pub/resources/software/ssaha2/")
99   versions_final <- NULL
100  for (url in urls) {
101    web <- read_html(url, encoding = "UTF-8")
102    files.txt <- web %>% html_nodes("p") %>% html_text()
103    files.row <- str_split(files.txt, "\n") %>% .[[1]] %>% as.character()
104    files.row <- str_split(files.row, " ")
105    files <- lapply(files.row, function(x) return(x[str_detect(x, "tgz"))])
106    files <- unlist(files)
107    versions <- str_extract(files, "_v.*tgz")
108    versions <- versions[!is.na(versions)]
109    versions <- str_replace_all(versions, "^_|.tgz$", "")
110    versions_final <- c(versions_final, versions)
111  }
112  return(versions_final)
113 }

```

```

114
115 # Function get fusioncatcher all versions
116 get.fusioncatcher.versions <- function() {
117   urls <- c("https://sourceforge.net/projects/fusioncatcher/files/")
118   versions_final <- NULL
119   for (url in urls) {
120     web <- read_html(url)
121     versions <- web %>% html_nodes("#files_list") %>% html_table() %>% .[[1]]
122     versions <- versions$Name
123     versions <- str_extract(versions, ".*[(tgz)(tar.gz)(zip)]+$")
124
125     versions <- versions[!is.na(versions)]
126     versions <- str_extract(versions, "v[0-9].*[pz]")
127     versions <- versions[!is.na(versions)]
128     versions <- str_replace(versions, "(.zip)|(.tgz)|(.tar.gz)|(.gz)", "")
129     versions_final <- c(versions_final, versions)
130   }
131   return(versions_final)
132 }
133
134 # Function get pigz all versions
135 get.pigz.versions <- function() {
136   urls <- c("http://cdn-fastly.deb.debian.org/debian/pool/main/p/pigz/")
137   versions_final <- NULL
138   for (url in urls) {
139     web <- read_html(url)
140     versions <- web %>% html_nodes("table") %>% html_table() %>% .[[1]]
141     versions <- versions$Name
142     versions <- str_extract(versions, ".*orig.*[(tgz)(tar.gz)(zip)]+$")
143
144     versions <- versions[!is.na(versions)]
145     versions <- str_extract(versions, "[0-9].*[pz]")
146     versions <- str_replace_all(versions, ".orig|(.zip)|(.tgz)|(.tar.gz)|(.gz)",
147       "")
148     versions_final <- c(versions_final, versions)
149   }
150   return(versions_final)
151 }
152
153 # Function get velvet all versions
154 get.velvet.versions <- function() {
155   urls <- c("http://www.ebi.ac.uk/~zerbino/velvet/")
156   versions_final <- NULL
157   for (url in urls) {
158     h <- basicTextGatherer()
159     web <- getURL(url, headerfunction = h$update)
160     web <- str_split(web, "\n")
161     web <- web[[1]]
162     web <- web[str_detect(web, "href")]
163     web <- web[str_detect(web, "velvet")]
164     web <- web[str_detect(web, "tgz")]
165     versions <- str_extract(web, "velvet_[0-9].*.tgz")
166     versions <- str_replace_all(versions, "velvet_|>|.tgz", "")
167     versions <- versions[!is.na(versions)]
168     versions_final <- c(versions_final, versions)
169   }
170   return(versions_final)
171 }
172

```

```

173 # Function get lzo all versions
174 get.lzo.versions <- function() {
175   urls <- c("http://www.oberhumer.com/opensource/lzo/download/")
176   versions_final <- NULL
177   for (url in urls) {
178     h <- basicTextGatherer()
179     web <- getURL(url, headerfunction = h$update)
180     web <- str_split(web, "\n")
181     web <- web[[1]]
182     web <- web[str_detect(web, "lzo")]
183     web <- web[str_detect(web, "tar.gz")]
184     web <- str_extract(web, ">lzo-[0-9].*.tar.gz")
185     versions <- str_replace_all(web, "lzo->|.tar.gz", "")
186     versions <- versions[!is.na(versions)]
187     versions_final <- c(versions_final, versions)
188   }
189   return(versions_final)
190 }
191
192 # Function get snpeff all versions
193 get.snpeff.versions <- function() {
194   urls <- c("https://sourceforge.net/projects/snpeff/files/")
195   versions_final <- NULL
196   for (url in urls) {
197     h <- basicTextGatherer()
198     web <- getURL(url, headerfunction = h$update)
199     web <- str_split(web, "\n")
200     web <- web[[1]]
201     web <- web[str_detect(web, "/download")]
202     web <- web[str_detect(web, "href")]
203     web <- web[!str_detect(web, "source=files")]
204     web <- str_extract(web, "v[0-9].*_core.zip")
205     versions <- str_replace(web, "(core.zip)|snpEff_", "")
206     versions <- versions[!is.na(versions)]
207     versions_final <- c(versions_final, versions)
208   }
209   return(versions_final)
210 }
211
212 # Function get zlib all versions
213 get.zlib.versions <- function() {
214   urls <- c("https://sourceforge.net/projects/libpng/files/zlib/")
215   versions_final <- NULL
216   for (url in urls) {
217     web <- read_html(url)
218     versions <- web %>% html_nodes("#files_list") %>% html_table() %>% .[[1]]
219     versions <- versions$Name
220     versions <- str_extract(versions, "[0-9.]*")
221
222     versions <- versions[!is.na(versions)]
223     versions <- versions[versions != ""]
224     versions_final <- c(versions_final, versions)
225   }
226   return(versions_final)
227 }
228
229 # Function get armadillo all versions
230 get.armadillo.versions <- function() {
231   urls <- c("https://sourceforge.net/projects/arma/files/")
232   versions_final <- NULL

```

```

233 for (url in urls) {
234   h <- basicTextGatherer()
235   web <- getURL(url, headerfunction = h$update)
236   web <- str_split(web, "\n")
237   web <- web[[1]]
238   web <- web[str_detect(web, "href")]
239   web <- web[!str_detect(web, "source=files")]
240   web <- web[str_detect(web, "files\\/armadillo")]
241   web <- web[!str_detect(web, "timeline")]
242   web <- str_extract(web, "[0-9][0-9.]*")
243   versions <- str_replace(web, ".$", "")
244   versions <- versions[!is.na(versions)]
245   versions_final <- c(versions_final, versions)
246 }
247 versions_final <- c(versions_final, "4.600.1")
248 return(versions_final)
249 }
250
251 # Function get prinseq all versions
252 get.prinseq.versions <- function() {
253   urls <- c("https://sourceforge.net/projects/prinseq/files/standalone/")
254   versions_final <- NULL
255   for (url in urls) {
256     h <- basicTextGatherer()
257     web <- getURL(url, headerfunction = h$update)
258     web <- str_split(web, "\n")
259     web <- web[[1]]
260     web <- web[str_detect(web, "href")]
261     web <- web[!str_detect(web, "source=files")]
262     web <- web[str_detect(web, "files/standalone/prinseq-lite")]
263     web <- web[!str_detect(web, "timeline")]
264     web <- str_extract(web, "[0-9][0-9.]*")
265     versions <- str_replace(web, ".$", "")
266     versions <- versions[!is.na(versions)]
267     versions_final <- c(versions_final, versions)
268   }
269   return(versions_final)
270 }
271
272 # Function get solexaqa all versions
273 get.solexaqa.versions <- function() {
274   urls <- c("https://sourceforge.net/projects/solexaqa/files/src/")
275   versions_final <- NULL
276   for (url in urls) {
277     h <- basicTextGatherer()
278     web <- getURL(url, headerfunction = h$update)
279     web <- str_split(web, "\n")
280     web <- web[[1]]
281     web <- web[str_detect(web, "href")]
282     web <- web[!str_detect(web, "source=files")]
283     web <- web[str_detect(web, "files/src/SolexaQA")]
284     web <- web[!str_detect(web, "timeline")]
285     web <- str_extract(web, "[v.][0-9][0-9.]*")
286     web <- str_replace(web, ".$", "")
287     versions <- str_replace(web, "^[v.]", "")
288     versions <- versions[!is.na(versions)]
289     versions_final <- c(versions_final, versions)
290   }
291   return(versions_final)
292 }

```

```

293
294 # Function get mapsplICE2 all versions
295 get.mapsplICE2.versions <- function() {
296   urls <- c("http://www.netlab.uky.edu/p/bioinfo/MapSplICE2")
297   versions_final <- NULL
298   for (url in urls) {
299     h <- basicTextGatherer()
300     web <- getURL(url, headerfunction = h$update)
301     web <- str_split(web, "\n")
302     web <- web[[1]]
303     web <- web[str_detect(web, "href")]
304     web <- web[str_detect(web, "download")]
305     versions <- str_extract(web, "[0-9]\\.[0-9.]*")
306     versions <- versions[!is.na(versions)]
307     versions_final <- c(versions_final, versions)
308   }
309   return(versions_final)
310 }
311
312 # Function get root all versions
313 get.root.versions <- function() {
314   urls <- c("https://root.cern.ch/releases")
315   versions_final <- NULL
316   for (url in urls) {
317     h <- basicTextGatherer()
318     web <- getURL(url, headerfunction = h$update)
319     web <- str_split(web, "\n")
320     web <- web[[1]]
321     web <- web[str_detect(web, "href")]
322     web <- web[str_detect(web, "Release")]
323     web <- str_extract(web, "[0-9]\\.[0-9./]*")
324     versions <- str_replace_all(web, fixed("/"), fixed("."))
325     versions <- versions[!is.na(versions)]
326     versions_final <- c(versions_final, versions)
327   }
328   versions_final <- sort(versions_final, decreasing = T)
329   return(versions_final)
330 }
331
332 # Function get curl all versions
333 get.curl.versions <- function() {
334   urls <- c("https://curl.haxx.se/download/")
335   versions_final <- NULL
336   for (url in urls) {
337     h <- basicTextGatherer()
338     web <- getURL(url, headerfunction = h$update)
339     web <- str_split(web, "\n")
340     web <- web[[1]]
341     web <- web[str_detect(web, "href")]
342     web <- web[str_detect(web, "curl-")]
343     web <- str_extract(web, "[0-9]\\.[0-9./]*")
344     web <- web[!is.na(web)]
345     versions <- web[!duplicated(web)]
346     versions <- str_replace(versions, ".$", "")
347     versions_final <- c(versions_final, versions)
348   }
349   versions_final <- sort(versions_final, decreasing = T)
350   return(versions_final)
351 }
352

```

```

353 # Function get r all versions
354 get.r.versions <- function() {
355   urls <- paste0("https://cran.r-project.org/src/base/R-", c(0, 1, 2, 3),
"/")
356   versions_final <- NULL
357   for (url in urls) {
358     h <- basicTextGatherer()
359     web <- getURL(url, headerfunction = h$update)
360     web <- str_split(web, "\n")
361     web <- web[[1]]
362     web <- web[str_detect(web, "href")]
363     web <- web[str_detect(web, "R-")]
364     web <- str_extract(web, "R.[0-9-a-z.]*.tgz|R.[0-9-a-z.]*.tar.gz")
365     web <- str_replace(web, ".tar.gz|.tgz$", "")
366     web <- str_replace(web, "R-", "")
367     versions <- web[!is.na(web)]
368     versions <- versions[!duplicated(versions)]
369     versions_final <- c(versions_final, versions)
370   }
371   versions_final <- sort(versions_final, decreasing = T)
372   return(versions_final)
373 }
374
375 get.pcre.versions <- function() {
376   urls <- "https://ftp.pcre.org/pub/pcre/"
377   versions_final <- NULL
378   for (url in urls) {
379     h <- basicTextGatherer()
380     web <- getURL(url, headerfunction = h$update)
381     web <- str_split(web, "\n")
382     web <- web[[1]]
383     web <- web[str_detect(web, "href")]
384     web <- web[str_detect(web, "pcre")]
385     web <- str_extract(web, "pcre[0-9-a-z.]*.tar.gz")
386     web <- web[!str_detect(web, "pcre2")]
387     web <- str_replace(web, ".tar.gz$", "")
388     web <- str_replace(web, "pcre-", "")
389     versions <- web[!is.na(web)]
390     versions <- versions[!duplicated(versions)]
391     versions_final <- c(versions_final, versions)
392   }
393   versions_final <- sort(versions_final, decreasing = T)
394   return(versions_final)
395 }
396
397 get.ensemble_grch38_reffa.versions <- function() {
398   urls <- "ftp://ftp.ensembl.org/pub/"
399   versions_final <- NULL
400   for (url in urls) {
401     web <- read_html(url, encoding = "UTF-8")
402     files.txt <- web %>% html_nodes("p") %>% html_text()
403     files.row <- str_split(files.txt, "\n") %>% .[[1]] %>% as.character()
404     files.row <- str_split(files.row, " ")
405     files <- lapply(files.row, function(x) return(x[str_detect(x,
"release"))])
406     files <- unlist(files)
407     versions <- str_extract(files, "[0-9][0-9]*")
408     versions <- versions[!is.na(versions)]
409     versions <- versions[versions > "75"]
410     versions <- unique(versions)

```

```

411     versions_final <- c(versions_final, versions)
412 }
413 versions_final <- sort(versions_final, decreasing = T)
414 return(versions_final)
415 }
416
417 get.ucsc_utils.versions <- function() {
418     urls <- "http://hgdownload.cse.ucsc.edu/admin/exe/"
419     versions_final <- NULL
420     for (url in urls) {
421         web <- read_html(url)
422         versions <- web %>% html_nodes("pre a") %>% html_attr(name = "href")
423         versions <- versions[str_detect(versions, "userApps")]
424         versions <- versions[str_detect(versions, ".v")]
425         versions <- str_extract(versions, "v[0-9]*")
426         versions <- versions[!is.na(versions)]
427         versions_final <- c(versions_final, versions)
428     }
429     versions_final <- sort(versions_final, decreasing = T)
430     return(versions_final)
431 }
432
433 get.sqlite.versions <- function() {
434     url <- "https://www.sqlite.org/chronology.html"
435     web <- read_html(url)
436     versions <- web %>% html_nodes("table") %>% html_table() %>% .[[1]]
437     versions <- versions$Version
438
439     pos <- str_split(versions, fixed("."))
440     func <- function(version.number) {
441         op <- options()
442         options(scipen = 200)
443
444         version.number <- as.numeric(version.number)
445         a <- version.number[1] * 1e+06
446         b <- version.number[2] * 10000
447         if (is.na(version.number[3])) {
448             result <- a + b
449             if (result < 3071506) {
450                 return(NULL)
451             } else {
452                 return(result)
453             }
454         } else {
455             c <- version.number[3] * 100
456         }
457         if (is.na(version.number[4])) {
458             result <- a + b + c
459         } else {
460             d <- version.number[4] * 1
461             result <- a + b + c + d
462         }
463         if (result < 3071506) {
464             return(NULL)
465         }
466         result <- as.character(result)
467         options(op)
468
469         return(result)
470     }

```



```

471 versions <- unlist(lapply(pos, func))
472 }
473
474 get.sratools.versions <- function() {
475   url <- "https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/"
476
477   web <- read_html(url)
478   versions <- web %>% html_nodes("pre a") %>% html_attr(name = "href")
479   versions <- versions[!str_detect(versions, "-prere")]
480   versions <- str_extract(versions, "[0-9-]*")
481   versions <- versions[versions != ""]
482
483   versions <- sort(versions, decreasing = T)
484 }
485
486 get.solexaqa.versions <- function() {
487   url <- "https://sourceforge.net/projects/solexaqa/files/src/"
488   web <- read_html(url)
489   versions <- web %>% html_nodes("#files_list") %>% html_table() %>% .[[1]]
490   versions <- versions$Name
491   versions <- str_extract(versions, "[0-9][.0-9]+[0-9]")
492   versions <- versions[!is.na(versions)]
493 }
494
495 get.reditools.versions <- function() {
496   urls <- c("https://sourceforge.net/projects/reditools/files/")
497   versions_final <- NULL
498   for (url in urls) {
499     web <- read_html(url)
500     versions <- web %>% html_nodes("#files_list") %>% html_table() %>% .[[1]]
501     versions <- versions$Name
502     versions <- str_extract(versions, ".*[(tgz)(tar.gz)(zip)]+$")
503
504     versions <- versions[!is.na(versions)]
505     versions <- str_extract(versions, "[0-9].*[pz]")
506     versions <- versions[!is.na(versions)]
507     versions <- str_replace(versions, "(.zip)|(.tgz)|(.tar.gz)|(.gz)", "")
508     versions_final <- c(versions_final, versions)
509   }
510   return(versions_final)
511 }
512
513 get.miniconda2.versions <- function() {
514   urls <- c("https://repo.continuum.io/miniconda/")
515   versions_final <- NULL
516   for (url in urls) {
517     web <- read_html(url)
518     versions <- web %>% html_table() %>% .[[1]]
519     versions <- versions$Filename
520     versions <- versions[str_detect(versions, "Miniconda2")]
521     versions <- str_replace(versions, "-Linux.*|-Windows.*|-MacOSX.*", "")
522     versions <- str_extract(versions, "-[0-9.(latest)]*")
523     versions <- unique(str_replace(versions, "^-", ""))
524     versions <- versions[!is.na(versions)]
525     versions_final <- c(versions_final, versions)
526   }
527   return(versions_final)
528 }
529
530 get.miniconda3.versions <- function() {

```

```

531 urls <- c("https://repo.continuum.io/miniconda/")
532 versions_final <- NULL
533 for (url in urls) {
534   web <- read_html(url)
535   versions <- web %>% html_table() %>% .[[1]]
536   versions <- versions$Filename
537   versions <- versions[str_detect(versions, "Miniconda3")]
538   versions <- str_replace(versions, "-Linux.*|-Windows.*|-MacOSX.*", "")
539   versions <- str_extract(versions, "-[0-9.(latest)]*")
540   versions <- unique(str_replace(versions, "^_", ""))
541   versions <- versions[!is.na(versions)]
542   versions_final <- c(versions_final, versions)
543 }
544 return(versions_final)
545 }
546
547 get.blast.versions <- function() {
548   urls <- "https://anaconda.org/bioconda/blast/files"
549   versions_final <- NULL
550   for (url in urls) {
551     web <- read_html(url, encoding = "UTF-8")
552     files.table <- web %>% html_nodes("table") %>% .[[1]] %>% html_table()
553     files <- files.table$Name
554     versions <- str_extract(files, "blast-[0-9]+[.0-9]*")
555     versions <- str_replace(versions, "blast-", "")
556     versions <- versions[!is.na(versions)]
557     versions_final <- c(versions_final, versions)
558   }
559   versions_final <- unique(versions_final)
560   versions_final <- sort(versions_final, decreasing = T)
561 }
562

```