Package 'cuRnet'

November 11, 2017

Type Package

Title cuRnet: an R package for graph traversing on GPU.

Version 0.5.2

Date 2017-11-11

Author Vincenzo Bonnici, Federico Busato, Stefano Aldegheri, Luciano Cascione, Alberto Arribas Carmena, Muodzhon Akhmedov, Francesco Bertoni, Nicola Bombieri, Ivo Kwee and Rosalba Giugno.

Maintainer Vincenzo Bonnici <vincenzo.bonnici@univr.it>

Description cuRnet provides a parallel implementaions of BFS (Breath-Firts Search), SCC (Strongly Connected Components) and SSSP (Single-Source Shortest Paths) developed on top of the CUDA framework. In future developments, more graph algorithms are planned to be included.

License MIT + file LICENSE

Depends R (>= 3.1.0), Rcpp

Imports Rcpp

LinkingTo Rcpp

RoxygenNote 6.0.1

R topics documented:

`cuRnet_bfs` *Breadth-first search. This function traverses the graph via a breadth-first search from a given set of source vertices, and returns depth of visited nodes.*

Description

Breadth-first search. This function traverses the graph via a breadth-first search from a given set of source vertices, and returns depth of visited nodes.

Usage

```
cuRnet_bfs(graph, sources)
```

Arguments

`graph`	A cuRnet graph object created with `cuRnet_graph`.
`sources`	The lists of source vertices from which to start BFSs. Per every source, one BFS is performed.

Value

A `NumericMatrix` having a number of rows equal to the number of source vertices, and a number of columns equal to the total number of vertices of th einput graph. Every row correspondo to a specific source vertex, and row cell reports the depth from the given source to the correspondig graph vertex.

Examples

```
## Not run:
library(igraph)
library(cuRnet)
rg <- sample_fitness_pl(100, 1000, 2.2, 2.3)
cdf <- data.frame( ends(rg, E(rg))[,1], ends(rg, E(rg))[,2] )
colnames(cdf) <- c("from", "to")
sources <- union(cdf$from, cdf$to)[1:20]
cg <- cuRnet_graph(cdf)
bfs <- cuRnet_bfs(cg, sources)
bfs[1,]

## End(Not run)
```

`cuRnet_graph` *Create a cuRnet graph object from a 3-column* `DataFrame`*. The* `DataFrame` *represents edges in the form (source vertex, destination vertex, weight). First two colums are of type* `charatcter`*, weights are of type* `numeric`*. Weights column is optional, but must be specified for algorithms that require the information, such as SSSP.*

Description

Create a cuRnet graph object from a 3-column `DataFrame`. The `DataFrame` represents edges in the form (source vertex, destination vertex, weight). First two colums are of type `charatcter`, weights are of type `numeric`. Weights column is optional, but must be specified for algorithms that require the information, such as SSSP.

Usage

```
cuRnet_graph(dataFrame)
```

Examples

```
## Not run:
library(STRINGdb)
library(igraph)
library(cuRnet)
ss <- STRINGdb$new( version="10", species=9606, score_threshold=900)
g <- ss$get_graph()
from <- unique(ends(g,E(g))[,1])[1:10]
x <- data.frame("from" = ends(g,E(g))[,1], "to" = ends(g,E(g))[,2], "score" = E(g)$combined_score/1000)
cgraph <- cuRnet_graph(x)

## End(Not run)
```

`cuRnet_scc` `cuRnet_scc`*: Strongly Connected Components This function computes strongly connected components membership for every vertex of the input graph.*

Description

`cuRnet_scc`: Strongly Connected Components This function computes strongly connected components membership for every vertex of the input graph.

Usage

```
cuRnet_scc(graph)
```

Arguments

`graph`	A cuRnet graph object created with `cuRnet_graph`.

Value

A `NumericMatrix` of 1 row and number of columns equal to the number of graph vertices. Each cell reports the identifier of the connected component associated with the corresponding vertex.

Examples

```
## Not run:
library(igraph)
library(cuRnet)
rg <- sample_fitness_pl(10000, 30000, 2.2, 2.3)
cdf <- data.frame( ends(rg, E(rg))[,1], ends(rg, E(rg))[,2] )
colnames(cdf) <- c("from", "to")
cg <- cuRnet_graph(cdf)
cc <- cuRnet_scc(cg)
length(unique(cc[1,])) #number of found strongly connected components

## End(Not run)
```

`cuRnet_sssp` *Single Source Shortest Paths: distances and predecessors. This function computes shortest paths from a series of vertices. For each source vertex, shortest paths to every vertex in the graph are computed. The function returns distances and predecessors.*

Description

Single Source Shortest Paths: distances and predecessors. This function computes shortest paths from a series of vertices. For each source vertex, shortest paths to every vertex in the graph are computed. The function returns distances and predecessors.

Usage

```
cuRnet_sssp(graph, from)
```

Arguments

`graph`	A cuRnet graph object created with `cuRnet_graph`.
`from`	A `CharacterVector` with the names of the source vertices.

Value

A list of two `NumericMatrix` indexed by "distances" and "predecessors". Rows are source vertices and colums are network vertices. A entry is: "distances" the distance along the shortest path from the source to the destination vertex; "predecessors" a minimal predecessor of the vertex and along the shortest path.

Examples

```
## Not run:
library(STRINGdb)
library(igraph)
library(cuRnet)
ss <- STRINGdb$new( version="10", species=9606, score_threshold=900)
g <- ss$get_graph()
from <- V(g)$name[1:10]
x <- data.frame("from" = ends(g,E(g))[,1], "to" = ends(g,E(g))[,2], "score" = E(g)$combined_score/1000)
cg <- cuRnet_graph(x)
ret <- cuRnet_sssp(g, from)
ret[["distances"]]
ret[["predecessors"]]

## End(Not run)
```

---

cuRnet_sssp_dists *Single Source Shortest Paths: distances only. This function computes shortest paths from a series of vertices. For each source vertex, shortest paths to every vertex in the graph are computed. The function returns only distances.*

Description

Single Source Shortest Paths: distances only. This function computes shortest paths from a series of vertices. For each source vertex, shortest paths to every vertex in the graph are computed. The function returns only distances.

Usage

```
cuRnet_sssp_dists(graph, from)
```

Arguments

`graph`	A cuRnet graph object created with `cuRnet_graph`.
`from`	A `CharacterVector` with the names of the source vertices.

Value

A `NumericMatrix` where rows are source vertices and colums are network vertices. An entry is the distance along the shortest path from the source to the destination vertex.

Examples

```
## Not run:
library(STRINGdb)
library(igraph)
library(cuRnet)
ss <- STRINGdb$new( version="10", species=9606, score_threshold=900)
g <- ss$get_graph()
from <- V(g)$name[1:10]
x <- data.frame("from" = ends(g,E(g))[,1], "to" = ends(g,E(g))[,2], "score" = E(g)$combined_score/1000)
cg <- cuRnet_graph(x)
ret <- cuRnet_sssp_dists(cg, from)
ret[["distances"]][1,]
ret[["predecessors"]][1,]
ret[1,]

## End(Not run)
```

Index