

## Install from source

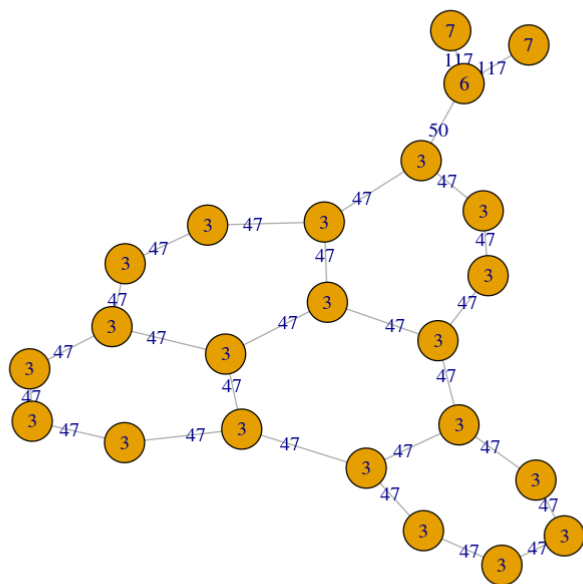
( develop branch is recommended to obtain feature vector representation of WL kernel)

```
$ git clone -b develop https://github.com/BorgwardtLab/graph-kernels.git  
$ cd graph-kernels/src/R  
$ R CMD build graphkernels  
$ R CMD INSTALL graphkernels_2.0.1.tar.gz
```

```
In [15]: library(graphkernels)
```

```
In [3]: data(mutag)
```

```
In [4]: plot(mutag[[1]])
```



## Compute WL kernel

```
In [7]: K <- CalculateWLKernel(mutag, 2, store.features = TRUE)
K$kernel[1:10, 1:10]
```

A matrix: 10 × 10 of type dbl

```
683 404 536 679 346 521 532 548 742 445
404 502 340 406 270 344 466 336 502 300
536 340 451 540 283 429 426 424 598 362
679 406 540 679 346 524 527 543 740 445
346 270 283 346 237 277 306 280 392 240
521 344 429 524 277 424 416 414 586 350
532 466 426 527 306 416 523 437 608 369
548 336 424 543 280 414 437 443 592 357
742 502 598 740 392 586 608 592 848 494
445 300 362 445 240 350 369 357 494 313
```

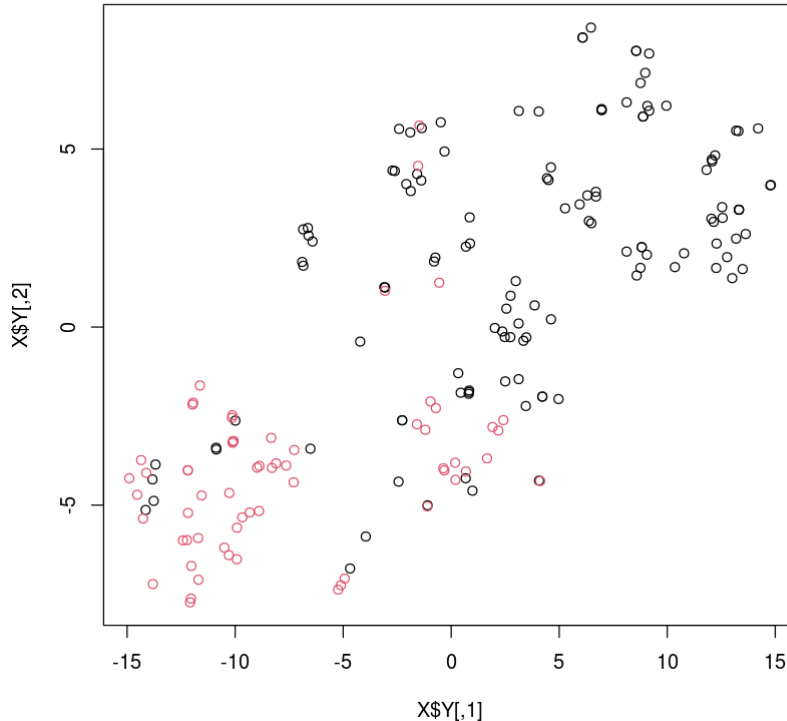
```
In [8]: head(K$feature)
```

A matrix: 6 × 214 of type int

1	2	3	4	5	6	7	8	9	10	...	206	207	208	209	210	211	212	213	214	215
0	0	20	0	0	1	2	0	0	0	...	0	0	0	0	0	0	0	0	0	2
0	0	13	0	0	4	9	0	0	0	...	0	1	0	0	0	0	0	0	0	8
0	0	16	0	0	1	2	0	0	0	...	0	0	0	0	0	0	0	0	0	2
0	0	20	0	0	1	2	0	0	0	...	0	0	0	0	0	0	0	0	0	2
0	0	12	0	0	3	2	0	0	0	...	0	0	0	0	0	0	0	0	0	2
0	0	16	0	0	1	3	0	0	0	...	0	1	0	0	0	0	0	0	0	2

## plot on 2D via tSNE

```
In [10]: ## install.packages("Rtsne")
library(Rtsne)
X <- Rtsne(K$feature, check_duplicates = FALSE)
y <- sapply(mutag, function(x) x$label)
plot(X$Y, col = y + 1)
```



## Perform classification by SVM

```
In [12]: ## install.packages("kernlab")
library(kernlab)
holdout <- c(which(y == 0)[1:10], which(y == 1)[1:10])
classifier <- ksvm(as.kernelMatrix(K$kernel[-holdout, -holdout]), y[-holdout], type="C-svc", kernel="matrix", scaling=FALSE, C=.01)
Ktest <- K$kernel[holdout, -holdout][,SVindex(classifier), drop=FALSE]
ypred <- predict(classifier, as.kernelMatrix(Ktest))
ypred == y[holdout]
```

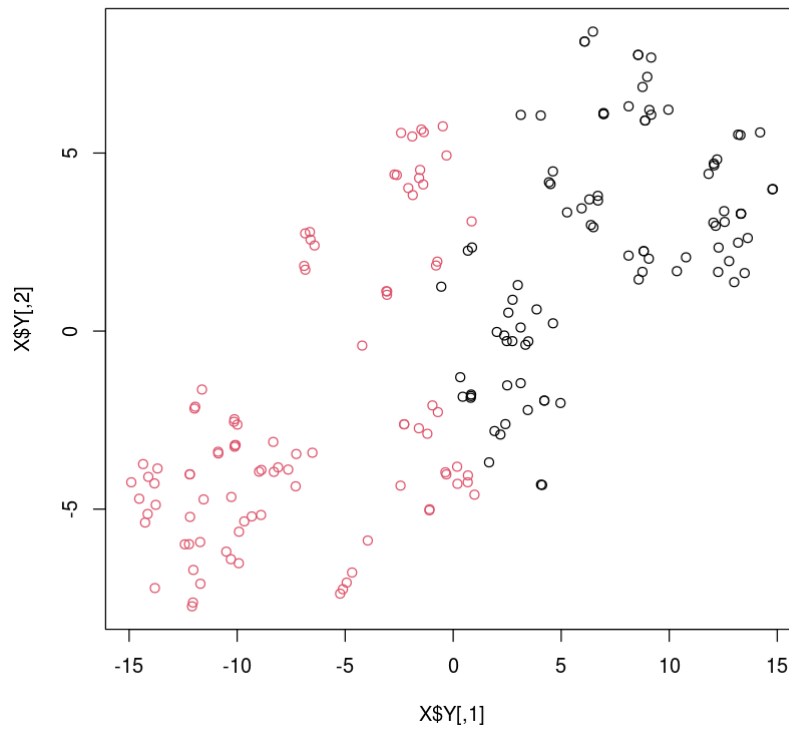
TRUE · TRUE · TRUE · TRUE · FALSE · TRUE · TRUE · TRUE · TRUE · TRUE ·  
FALSE · FALSE · TRUE · TRUE · FALSE · TRUE · FALSE · TRUE · FALSE · TRUE

```
In [13]: sum(ypred == y[holdout]) / length(holdout)
```

0.7

## Perform clustering by kernel k-means

```
In [14]: c12 <- kkmeans(as.kernelMatrix(K$kernel), centers = 2)
plot(X$Y, col = c12@.Data)
```



```
In [ ]:
```