COG-56 RNA velocity using scvelociraptor

- This is a benchmark for applying RNA velocity to CogentDS.
    - In the current CogentDS, we use the Bioconductor SingleCellExperiment to deal with single-cell matrix data(https://github.com/takarabiousa/CogentDS/blob/master/R/CogentDS.R).
    - The one potential R package to overlay on CogentDS: velociraptor (https://kevinrue.github.io/velociraptor/reference/scvelo.html)
    - This package provides a lightweight interface between the Bioconductor SingleCellExperiment data structure and the scvelo Python package for RNA velocity calculations.
    - This function uses the scVelo Python package (https://pypi.org/project/scvelo/) for RNA velocity calculations. The main difference from the original velocyto approach is that the dynamical model of scVelo does not rely on the presence of observed steady-state populations, which should improve the reliability of the velocity calculations in general applications.
- Need to create SingleCellExperiment structure dataset from matrix data.
    - Reference: https://robertamezquita.github.io/orchestratingSingleCellAnalysis/data-infrastructure.html
- (FYI) R package list in CogentDS

```
Imports:
  umap (>= 0.2.2.0),
  DT (>= 0.7),
  spatstat (<= 1.64-1),
  Seurat (>= 3.0.2),
  RANN (>= 2.6.1),
  Matrix (>= 1.2-17),
  data.table (>= 1.12.2),
  cluster (>= 2.1.0),
  plotly (>= 4.8.0),
  pheatmap (>= 1.0.12),
  sm (>= 2.2-5.4),
  pkgmaker (>= 0.25.8),
  shinyBS (>= 0.61),
  shinythemes (>= 1.1.1),
  colourpicker (>= 1.0),
  crosstalk (>= 1.0.0),
  shinycssloaders (>= 0.2.0),
  optparse (>= 1.4.4),
  reshape2 (>= 1.4.3),
  lattice (>= 0.20-35),
  RColorBrewer (>= 1.1-2),
  Rtsne (>= 0.13),
```

```
  gridExtra (>= 2.3),
  gtable (>= 0.2.0),
  naturalsort (>= 0.1.3),
  futile.logger (>= 1.4.3),
  knitr (>= 1.22),
  kableExtra (>= 0.7.0),
  plotrix (>= 3.7),
  shiny (>= 1.4.0),
  shinyjs (>= 1.0),
  rmarkdown (>= 1.9),
  htmltools (>= 0.3.6),
  methods (>= 3.6.0),
  multtest (>= 2.4),
  SingleCellExperiment (>= 1.10.1),
```

Bechmarking RNA velocity in a Bioconductor framework based on tutorial

- Server: davinci
- R installed by anaconda
- Confirm this R package works as expected
- Tutorial: https://kevinrue.github.io/velociraptor/articles/velociraptor.html

Install R package based on CogentDS

About R environment

```
(base) [tuchiyama@davinci tools]$ /wgbs/scratch/tuchiyama/anaconda3/bin/R

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-conda-linux-gnu (64-bit)
```

Install scater

- R package 'scater' generates PCA, UMAP or TSNE plot.
- This package is compatible with the bioconductor SingleCellExperiment, which we use to deal with single-cell matrix in current CogentDS.
- install scater via conda because there can be issues when installing R via conda and a package via R based on Mike's knowledge. Thank you, Mike!

```
(base) [tuchiyama@davinci tuchiyama]$ conda install -c bioconda
bioconductor-scater

> library(scater)
Loading required package: SingleCellExperiment
Loading required package: SummarizedExperiment
```

```
Loading required package: MatrixGenerics
Loading required package: matrixStats
```

Install other packages

```
install.packages('BiocManager')
BiocManager::install('velociraptor', dependencies = TRUE)
BiocManager::install(c('scRNAseq', 'SingleCellExperiment', 'scuttle',
'scran', 'Rtsne', 'umap'))
install.packages('Matrix')
```

(1) Load demo data

- use package 'scRNAseq' to load in order to understand velociraptor efficiently based on tutorial
- However, loading demo data was failed, so the data frame for SingleCellExperiment was created manually. In this case, 11,000 cells in A549 was used.

Load demo data by scRNAseq

```
> library(scRNAseq)
Loading required package: SingleCellExperiment
Loading required package: SummarizedExperiment
Loading required package: MatrixGenerics
Loading required package: matrixStats

> sce <- HermannSpermatogenesisData()
snapshotDate(): 2022-10-31
see ?scRNAseq and browseVignettes('scRNAseq') for documentation
loading from cache
Error: failed to load resource
  name: EH3469
  title: Hermann Spermatogenesis spliced counts
  reason: error reading from connection
> sce
Error: object 'sce' not found
```

Create data frame for "SingleCellExperiment"

```
library(Matrix)
library(SingleCellExperiment)

gene_mtx <-
readMM('/wgbs_global1/davinci/tuchiyama/A549/output/genes.mtx.gz')
barcode <-
read.delim('/wgbs_global1/davinci/tuchiyama/A549/output/barcodes.tsv',
header=FALSE)
gene <-
```

```
read.delim('/wgbs_global1/davinci/tuchiyama/A549/output/genes.tsv',
header=FALSE)
gene_mtx <- t(gene_mtx)
colnames(gene_mtx) <- barcode$V1
rownames(gene_mtx) <- gene$V1

# Reference:
https://robertamezquita.github.io/orchestratingSingleCellAnalysis/data-
infrastructure.html
# current CogentDS code:
https://github.com/takarabiousa/CogentDS/blob/master/R/CogentDS.R
sce <- SingleCellExperiment(
  assays = list(
    counts=as(gene_mtx, 'dgCMatrix') # for matrix type, same as current
CogentDS
  ),
  colData=barcode,
  rowData=gene
)

intron_mtx <-
readMM('/wgbs_global1/davinci/tuchiyama/A549/output/introns.mtx.gz')
intron <-
read.delim('/wgbs_global1/davinci/tuchiyama/A549/output/introns.tsv',
header=FALSE)
intron_mtx <- t(intron_mtx)
colnames(intron_mtx) <- barcode$V1
rownames(intron_mtx) <- gene$V1

assay(sce, 'spliced') <- gene_mtx
assay(sce, 'unspliced') <- intron_mtx


"""
# Just in case, filtering based on tsv file
file <-
'/wgbs_global1/davinci/tuchiyama/index/Homo_sapiens.GRCh38.94.expanded.fea
tures.tsv'
splitDf <- read.delim(file, header=TRUE, as.is=TRUE)
colnames(splitDf)[colnames(splitDf) == 'intron'] <- 'unspliced'

#  need to deal with the following process as proprocessing by
prep_scVelo.py
gene_df <- as.data.frame(as.matrix(gene_mtx))
assay(sce, 'spliced') <- as(gene_df[splitDf$spliced, ], 'sparseMatrix')

intron_df <- as.data.frame(as.matrix(intron_mtx))
intron_sel <- intron_df[splitDf$unspliced, ]
rownames(intron_sel) <- splitDf$spliced
assay(sce, 'unspliced') <- as(intron_sel, 'sparseMatrix')
"""
```

Check output

```
> sce
class: SingleCellExperiment
dim: 58735 11000
metadata(0):
assays(3): counts spliced unspliced
rownames(58735): ENSG00000223972 ENSG00000243485 ... ENSG00000278625
  ENSG00000277374
rowData names(2): V1 V2
colnames(11000): AACCGGTTAACTCCGGACTTACGT AACCGGTTAACTTAACAGCGGCAA ...
  TGGATCAATTGACTATGGTCAGAT TGGATCAATTGGAGAGAATTCGGT
colData names(1): V1
reducedDimNames(0):
mainExpName: NULL
altExpNames(0):
```

(FYI) Measure the time for data loading

- The time for loding sparse matrix designed to Coordinate format was 5 times faster than that for raw matrix data(~genematrix.csv), which is default output in current CogentAP
- Both of them are 11,000 cells x 58735 genes matrix file from A549 cell line.

```
system.time(read.csv('/wgbs/scratch2/hanbunathan/BIS/SKY-
30/analyze_hg38/analyze_hg38_genematrix.csv', row.names=1))
   user  system elapsed
131.830   4.842 136.999

system.time(readMM('/wgbs_global1/davinci/tuchiyama/A549/output/genes.mtx.
gz'))
   user  system elapsed
33.496   1.792  35.372
```

(2) Basic workflow using velociraptor

- Failed to download miniconda, so the following process could not go.

```
> library(scuttle)
> sce <- logNormCounts(sce, assay.type=1)

> library(scran)
> dec <- modelGeneVar(sce)
> top.hvgs <- getTopHVGs(dec, n=2000)

> library(velociraptor)
Registered S3 methods overwritten by 'zellkonverter':
  method                                          from
  py_to_r.numpy.ndarray                           reticulate
  py_to_r.pandas.core.arrays.categorical.Categorical reticulate
```

```
> velo.out <- scvelo(sce, subset.row=top.hvgs, assay.X="spliced")

trying URL 'https://repo.anaconda.com/miniconda/Miniconda3-py38_4.12.0-
Linux-x86_64.sh'
Content type 'application/x-sh' length 76120962 bytes (72.6 MB)

downloaded 183 KB

Error in download.file(url, fname, mode = "wb") :
  download from 'https://repo.anaconda.com/miniconda/Miniconda3-
py38_4.12.0-Linux-x86_64.sh' failed
In addition: Warning messages:
1: In download.file(url, fname, mode = "wb") :
  downloaded length 188128 != reported length 76120962
2: In download.file(url, fname, mode = "wb") :
  URL 'https://repo.anaconda.com/miniconda/Miniconda3-py38_4.12.0-Linux-
x86_64.sh': status was 'Failed writing received data to disk/application'

> velo.out
Error: object 'velo.out' not found
```

Benchmark on local MacOS enviroment

- This benchmerk cannot be conducted on davinci, but can on local MacOS.
- There can be issues related to network connection in davincibecause library 'scRNAseq' downloaded demo data when it was used. Moreover, velociraptor automatically installed miniconda package.

(1) Load demo data

```
> library(scRNAseq)
Loading required package: SingleCellExperiment
Loading required package: SummarizedExperiment
Loading required package: MatrixGenerics
Loading required package: matrixStats

> sce <- HermannSpermatogenesisData()
/Users/TomoyaUchiyama/Library/Caches/org.R-project.R/R/ExperimentHub
  does not exist, create directory? (yes/no): yes
  |======================================================================|
100%

snapshotDate(): 2023-04-24
see ?scRNAseq and browseVignettes('scRNAseq') for documentation
downloading 1 resources
retrieving 1 resource
  |======================================================================|
100%
```

```
loading from cache
see ?scRNAseq and browseVignettes('scRNAseq') for documentation
downloading 1 resources
retrieving 1 resource
  |======================================================================|
100%

loading from cache
see ?scRNAseq and browseVignettes('scRNAseq') for documentation
downloading 1 resources
retrieving 1 resource
  |======================================================================|
100%

> sce
class: SingleCellExperiment
dim: 54448 2325
metadata(0):
assays(2): spliced unspliced
rownames(54448): ENSMUSG00000102693.1 ENSMUSG00000064842.1 ...
  ENSMUSG00000064369.1 ENSMUSG00000064372.1
rowData names(0):
colnames(2325): CCCATACTCCGAAGAG AATCCAGTCATCTGCC ... ATCCACCCACCACCAG
  ATTGGTGGTTACCGAT
colData names(1): celltype
reducedDimNames(0):
mainExpName: NULL
altExpNames(0):
```

(2) Basic workflow using velociraptor

```
# Downsampling for demonstration
> sce <- sce[, 1:500]

> library(scuttle)
Warning message:
package 'scuttle' was built under R version 4.3.1
> sce <- logNormCounts(sce, assay.type=1)

> library(scran)
Warning message:
package 'scran' was built under R version 4.3.1
> dec <- modelGeneVar(sce)
Warning message:
In regularize.values(x, y, ties, missing(ties), na.rm = na.rm) :
  collapsing to unique 'x' values
> top.hvgs <- getTopHVGs(dec, n=2000)

> library(velociraptor)
Registered S3 methods overwritten by 'zellkonverter':
  method                                           from
  py_to_r.numpy.ndarray                            reticulate
```

```
  py_to_r.pandas.core.arrays.categorical.Categorical reticulate

> velo.out <- scvelo(sce, subset.row=top.hvgs, assay.X="spliced")
trying URL 'https://repo.anaconda.com/miniconda/Miniconda3-py38_4.12.0-
MacOSX-x86_64.sh'
Content type 'application/x-sh' length 59150526 bytes (56.4 MB)
==================================================
downloaded 56.4 MB

PREFIX=/Users/TomoyaUchiyama/Library/Caches/org.R-
project.R/R/basilisk/1.12.1/0
Unpacking payload ...
Collecting package metadata (current_repodata.json): done
Solving environment: done

computing neighbors
OMP: Info #276: omp_set_nested routine deprecated, please use
omp_set_max_active_levels instead.
    finished (0:00:05) --> added
    'distances' and 'connectivities', weighted adjacency matrices
(adata.obsp)
computing moments based on connectivities
    finished (0:00:00) --> added
    'Ms' and 'Mu', moments of un/spliced abundances (adata.layers)
computing velocities
    finished (0:00:00) --> added
    'velocity', velocity vectors for each individual cell (adata.layers)
computing velocity graph
    finished (0:00:00) --> added
    'velocity_graph', sparse matrix with cosine correlations (adata.uns)
computing terminal states
    identified 1 region of root cells and 1 region of end points .
    finished (0:00:00) --> added
    'root_cells', root cells of Markov diffusion process (adata.obs)
    'end_points', end points of Markov diffusion process (adata.obs)
--> added 'velocity_length' (adata.obs)
--> added 'velocity_confidence' (adata.obs)
--> added 'velocity_confidence_transition' (adata.obs)

> velo.out
class: SingleCellExperiment
dim: 2000 500
metadata(4): neighbors velocity_params velocity_graph velocity_graph_neg
assays(6): X spliced ... Mu velocity
rownames(2000): ENSMUSG00000117819.1 ENSMUSG00000081984.3 ...
ENSMUSG00000022965.8 ENSMUSG00000094660.2
rowData names(3): velocity_gamma velocity_r2 velocity_genes
colnames(500): CCCATACTCCGAAGAG AATCCAGTCATCTGCC ... CACCTTGTCGTAGGAG
TTCCCAGAGACTAAGT
colData names(7): velocity_self_transition root_cells ...
velocity_confidence velocity_confidence_transition
reducedDimNames(1): X_pca
mainExpName: NULL
altExpNames(0):
```

```
> library(scater)
Loading required package: ggplot2

> set.seed(100)
> sce <- runPCA(sce, subset_row=top.hvgs)
> sce <- runTSNE(sce, dimred="PCA")
> sce$velocity_pseudotime <- velo.out$velocity_pseudotime
> plotTSNE(sce, colour_by="velocity_pseudotime")
> embedded <- embedVelocity(reducedDim(sce, "TSNE"), velo.out)
ℹ Using the 'X' assay as the X matrix
computing velocity embedding
    finished (0:00:00) --> added
    'velocity_target', embedded velocity vectors (adata.obsm)
> grid.df <- gridVectors(sce, embedded, use.dimred = "TSNE")
```

(3) Visulaization

```
> library(ggplot2)
> plotTSNE(sce, colour_by="velocity_pseudotime") +
+     geom_segment(data=grid.df, mapping=aes(x=start.1, y=start.2,
+                                            xend=end.1, yend=end.2,
colour=NULL), arrow=arrow(length=unit(0.05, "inches")))

# check output
> sce
class: SingleCellExperiment
dim: 54448 500
metadata(0):
assays(3): spliced unspliced logcounts
rownames(54448): ENSMUSG00000102693.1 ENSMUSG00000064842.1 ...
ENSMUSG00000064369.1 ENSMUSG00000064372.1
rowData names(0):
colnames(500): CCCATACTCCGAAGAG AATCCAGTCATCTGCC ... CACCTTGTCGTAGGAG
TTCCCAGAGACTAAGT
colData names(3): celltype sizeFactor velocity_pseudotime
reducedDimNames(2): PCA TSNE
mainExpName: NULL
altExpNames(0):
```

RNA velocity coupled with pseudotime using A549 cell line

- 500 cells

```
library(Matrix)
library(SingleCellExperiment)
library(scuttle)
library(scran)
```

```
library(velociraptor)
library(scater)
library(ggplot2)

gene_mtx <- readMM('/Users/TomoyaUchiyama/WORKSPACE/salmon/genes.mtx.gz')
barcode <-
read.delim('/Users/TomoyaUchiyama/WORKSPACE/salmon/barcodes.tsv',
header=FALSE)
gene <- read.delim('/Users/TomoyaUchiyama/WORKSPACE/salmon/genes.tsv',
header=FALSE)
gene_mtx <- t(gene_mtx)
colnames(gene_mtx) <- barcode$V1
rownames(gene_mtx) <- gene$V1

sce <- SingleCellExperiment(assays = list(counts=as(gene_mtx,
'dgCMatrix')), colData=barcode, rowData=gene)

intron_mtx <-
readMM('/Users/TomoyaUchiyama/WORKSPACE/salmon/introns.mtx.gz')
intron_mtx <- t(intron_mtx)
colnames(intron_mtx) <- barcode$V1
rownames(intron_mtx) <- gene$V1

assay(sce, 'spliced') <- gene_mtx
assay(sce, 'unspliced') <- intron_mtx

# Downsampling for demonstration
sce <- sce[, 1:500]
sce <- logNormCounts(sce, assay.type=1)
dec <- modelGeneVar(sce)
top.hvgs <- getTopHVGs(dec, n=2000)
velo.out <- scvelo(sce, subset.row=top.hvgs, assay.X="spliced")

> velo.out
class: SingleCellExperiment
dim: 2000 500
metadata(4): neighbors velocity_params velocity_graph velocity_graph_neg
assays(6): X spliced ... Mu velocity
rownames(2000): ENSG00000038427 ENSG00000117724 ... ENSG00000132967
ENSG00000068985
rowData names(3): velocity_gamma velocity_r2 velocity_genes
colnames(500): AACCGGTTAACTCCGGACTTACGT AACCGGTTAACTTAACAGCGGCAA ...
AGAGTTCTAATTACCAAACTCCGA AGAGTTCTAATTACCAACCTTATT
colData names(7): velocity_self_transition root_cells ...
velocity_confidence velocity_confidence_transition
reducedDimNames(1): X_pca
mainExpName: NULL
altExpNames(0):

set.seed(100)
sce <- runPCA(sce, subset_row=top.hvgs)
sce <- runTSNE(sce, dimred="PCA")
sce$velocity_pseudotime <- velo.out$velocity_pseudotime
plotTSNE(sce, colour_by="velocity_pseudotime")
```

```
embedded <- embedVelocity(reducedDim(sce, "TSNE"), velo.out)
grid.df <- gridVectors(sce, embedded, use.dimred = "TSNE")
plotTSNE(sce, colour_by="velocity_pseudotime") +
+     geom_segment(data=grid.df, mapping=aes(x=start.1, y=start.2,
+                                            xend=end.1, yend=end.2,
colour=NULL), arrow=arrow(length=unit(0.05, "inches")))
```

- All cells (11,000 cells)

```
library(Matrix)
library(SingleCellExperiment)
library(scuttle)
library(scran)
library(velociraptor)
library(scater)
library(ggplot2)

gene_mtx <- readMM('/Users/TomoyaUchiyama/WORKSPACE/salmon/genes.mtx.gz')
barcode <-
read.delim('/Users/TomoyaUchiyama/WORKSPACE/salmon/barcodes.tsv',
header=FALSE)
#barcode <- data.frame(barcode[order(barcode$V1), ])
colnames(barcode) <- 'cell'
gene <- read.delim('/Users/TomoyaUchiyama/WORKSPACE/salmon/genes.tsv',
header=TRUE)
gene_mtx <- t(gene_mtx)
colnames(gene_mtx) <- barcode$cell
rownames(gene_mtx) <- gene$gene_name

sce <- SingleCellExperiment(assays = list(counts=as(gene_mtx,
'dgCMatrix')), colData=barcode, rowData=gene)

#'as(<dgTMatrix>, "dgCMatrix")' is deprecated.
#Use 'as(., "CsparseMatrix")' instead.
#See help("Deprecated") and help("Matrix-deprecated").

intron_mtx <-
readMM('/Users/TomoyaUchiyama/WORKSPACE/salmon/introns.mtx.gz')
intron_mtx <- t(intron_mtx)
colnames(intron_mtx) <- barcode$cell
rownames(intron_mtx) <- gene$gene_name
assay(sce, 'spliced') <- gene_mtx
assay(sce, 'unspliced') <- intron_mtx

cellgroup <-
'/Users/TomoyaUchiyama/WORKSPACE/salmon/metadata_sample_info.csv'
df_cellgroup <- read.csv(cellgroup)
df_cellgroup <- df_cellgroup[order(df_cellgroup$Barcode), ]
rownames(df_cellgroup) <- 1:nrow(df_cellgroup)
sce$cell_group <- df_cellgroup$BC1

set.seed(100)
```

```
sce <- logNormCounts(sce, assay.type=1)
dec <- modelGeneVar(sce)
Warning message:
In regularize.values(x, y, ties, missing(ties), na.rm = na.rm) :
  collapsing to unique 'x' values
top.hvgs <- getTopHVGs(dec, n=2000)

velo.out <- scvelo(sce, subset.row=top.hvgs, assay.X="spliced")

> velo.out
class: SingleCellExperiment
dim: 2000 11000
metadata(4): neighbors velocity_params velocity_graph velocity_graph_neg
assays(6): X spliced ... Mu velocity
rownames(2000): ENSG00000038427 ENSG00000274012 ... ENSG00000213949
ENSG00000103034
rowData names(3): velocity_gamma velocity_r2 velocity_genes
colnames(11000): AACCGGTTAACTCCGGACTTACGT AACCGGTTAACTCCGGAGTATAGT ...
TGGATCAATTGACTATGGTCAGAT
  TGGATCAATTGGAGAGAATTCGGT
colData names(7): velocity_self_transition root_cells ...
velocity_confidence
  velocity_confidence_transition
reducedDimNames(1): UMAP
mainExpName: NULL
altExpNames(0):

sce <- runPCA(sce, subset_row=top.hvgs)
sce <- runUMAP(sce, dimred="PCA")

# velocity_pseudotime by UMAP ---
sce$velocity_pseudotime <- velo.out$velocity_pseudotime
plotUMAP(sce, colour_by="velocity_pseudotime") +
scale_color_gradientn(name='pseudotime', colours =
c("#a50026","#d73027","#f46d43","#fdae61","#fee090", "#ffffbf", "#e0f3f8",
"#abd9e9", "#74add1", "#4575b4", "#4169E1"), values = c(1.0, 0.9, 0.8,
0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0))

embedded <- embedVelocity(reducedDim(sce, "UMAP"), velo.out)
grid.df <- gridVectors(sce, embedded, use.dimred = "UMAP")
p1 <- plotUMAP(sce, colour_by="velocity_pseudotime") +
scale_color_gradientn(colours =
c("#a50026","#d73027","#f46d43","#fdae61","#fee090", "#ffffbf", "#e0f3f8",
"#abd9e9", "#74add1", "#4575b4", "#4169E1"), values = c(1.0, 0.9, 0.8,
0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0)) + geom_segment(data=grid.df,
mapping=aes(x=start.1, y=start.2, xend=end.1, yend=end.2, colour=NULL),
arrow=arrow(length=unit(0.05, "inches")))

# sample type by UMAP ---
plotUMAP(sce, colour_by="cell_group")

embedded <- embedVelocity(reducedDim(sce, "UMAP"), velo.out)
grid.df <- gridVectors(sce, embedded, use.dimred = "UMAP")
```

```
p2 <- plotUMAP(sce, colour_by="cell_group") + geom_segment(data=grid.df,
mapping=aes(x=start.1, y=start.2, xend=end.1, yend=end.2, colour=NULL),
arrow=arrow(length=unit(0.05, "inches")))

# gene ---
install.packages('Seurat')
library(Seurat)

velo.out <- scvelo(sce, subset.row=top.hvgs, assay.X="spliced",
use.dimred="UMAP")
reducedDimNames(velo.out) <- "UMAP"

p3 <- plotVelocity(velo.out, c("ENSG00000038427","ENSG00000274012"))
CombinePlots(plots = list(p1, p2, p3))
Warning messages:
1: CombinePlots is being deprecated. Plots should now be combined using
the patchwork system.
2: Graphs cannot be vertically aligned unless the axis parameter is set.
Placing graphs unaligned.
3: Graphs cannot be horizontally aligned unless the axis parameter is set.
Placing graphs unaligned.
```