

# Working with collections of communities (0.1-639)

Lawrence Hudson

2024-07-24

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Datasets</b>	<b>1</b>
<b>3</b>	<b>Community collection representation</b>	<b>2</b>
3.1	Basic operations . . . . .	2
3.2	Subsets . . . . .	3
3.3	Community properties . . . . .	4
3.4	Node properties . . . . .	10
3.5	Trophic link properties . . . . .	11
<b>4</b>	<b>Plotting</b>	<b>13</b>
4.1	Plot-per-community . . . . .	13
<b>5</b>	<b>Modifying communities</b>	<b>16</b>
<b>6</b>	<b>Ordering collections</b>	<b>19</b>
<b>7</b>	<b>Aggregating communities</b>	<b>21</b>
<b>8</b>	<b>‘Global’ node IDs</b>	<b>25</b>
8.1	Create IDs . . . . .	25
8.2	Table of properties . . . . .	25
8.3	Plot IDs . . . . .	28

## 1 Introduction

Cheddar provides functions for managing collections of communities, allowing you to perform inter-web comparisons such as examining changes in community structure over environmental, temporal and spatial gradients. You should read the ‘CheddarQuickstart’ and ‘Community’ vignettes before reading this one. The ‘ImportExport’ vignette shows how to get collections of communities in to Cheddar.

## 2 Datasets

Cheddar contains some published empirical food web collection datasets (Table 1).

Community	Notes	References
Millstream	The control and drought treatments from one replicate of a long-running study investigating how drought affects community structure	Ledger et al. (2011) Ledger et al. (2012) Woodward et al. (2012)
pHWebs	Ten of the twenty stream communities sampled across a wide pH gradient	Layer et al. (2010)

Table 1: Community collection data in Cheddar

### 3 Community collection representation

#### 3.1 Basic operations

Cheddar's `CommunityCollection` is a sub-class of R's `list`.

```
> data(pHWebs)
> pHWebs
```

A collection of 10 communities

Each element in a `CommunityCollection` is a `Cheddar Community`. Many of the usual `list` operations can be used.

```
> length(pHWebs)
```

```
[1] 10
```

```
> is.list(pHWebs)
```

```
[1] TRUE
```

```
> names(pHWebs)
```

```
[1] "Old Lodge"          "Afon Hafren"        "Broadstone"         "Dargall Lane"
[5] "Mosedal Beck"      "Duddon Pike Beck"  "Allt a'Mharcaidh"  "Hardknott Gill"
[9] "Bere Stream"       "Mill Stream"
```

```
> # Access first community in the collection
```

```
> pHWebs[[1]]
```

Old Lodge containing 23 nodes and 137 trophic links

```
> # Access a community by name
```

```
> pHWebs[['Broadstone']]
```

Broadstone containing 25 nodes and 178 trophic links

```
> # The number of trophic links in Broadstone
```

```
> NumberOfTrophicLinks(pHWebs[['Broadstone']])
```

```
[1] 178
```

```
> # The number of trophic links in each of the ten webs
> sapply(pHWebs, 'NumberOfTrophicLinks')
```

Old Lodge	Afon Hafren	Broadstone	Dargall Lane	Mosedal Beck
137	135	178	99	108
Duddon Pike Beck	Allt a'Mharcaidh	Hardknott Gill	Bere Stream	Mill Stream
286	334	386	943	1654

In contrast to R's `lists`, you can't change collections directly. This is because many checks are enforced when community collection objects are created, so you can not, for example, modify a collection's length or insert values in to the collection. The following operations would raise errors if executed.

```
> length(pHWebs) <- 2 # You can't do this
> pHWebs[1] <- "This will not work"
```

`CommunityCollection` guarantees that the title of each `Community` will be unique within a collection. The following will therefore always be `TRUE`.

```
> all(FALSE==duplicated(names(pHWebs)))
```

```
[1] TRUE
```

If the `Community` objects within a collection have body mass, `CommunityCollection` also guarantees that they will have the same units, as given in the community property `'M.units'`. Similarly, all communities in a collection will have the same `'N.units'`, if they contain numerical abundance data.

### 3.2 Subsets

You can use list operators to take subsets of collections or to reorder them.

```
> # Returns a new CommunityCollection that contains every other web
> pHWebs[seq(1, 10, by=2)]
```

A collection of 5 communities

```
> # Returns a new CommunityCollection with the order reversed
> pHWebs[10:1]
```

A collection of 10 communities

```
> # Returns a new CommunityCollection containing only these two webs
> pHWebs[c('Old Lodge', 'Bere Stream')]
```

A collection of 2 communities

### 3.3 Community properties

The `CollectionCPS` (for **Collection Community Properties**) returns a `data.frame` of properties.

```
> CollectionCPS(pHWebs)
```

	title	M.units	N.units	code	pH	lat	long
Old Lodge	Old Lodge	mg	m <sup>2</sup>	OLD	5.0	51.04	0.080
Afon Hafren	Afon Hafren	mg	m <sup>2</sup>	HAF	5.3	52.47	-3.700
Broadstone	Broadstone	mg	m <sup>2</sup>	BRO	5.5	51.08	0.053
Dargall Lane	Dargall Lane	mg	m <sup>2</sup>	DAR	5.8	55.08	-4.430
Mosedal Beck	Mosedal Beck	mg	m <sup>2</sup>	DUD3	5.9	54.41	-3.140
Duddon Pike Beck	Duddon Pike Beck	mg	m <sup>2</sup>	DUD1	6.1	54.41	-3.170
Allt a'Mharcaidh	Allt a'Mharcaidh	mg	m <sup>2</sup>	MHA	6.5	57.12	-3.850
Hardknott Gill	Hardknott Gill	mg	m <sup>2</sup>	DUD2	7.0	54.40	-3.170
Bere Stream	Bere Stream	mg	m <sup>2</sup>	BER	7.5	50.73	-2.210
Mill Stream	Mill Stream	mg	m <sup>2</sup>	MIL	8.4	50.68	-2.180

The table above shows all ‘first-class’ properties in all of the contained communities. `CommunityCollection` places no restrictions on first-class properties such as pH - it is possible for a `Community` within a collection to not have the pH property, to have a pH of `NA` or even to have an invalid pH, for example a negative value.

`CollectionCPS` takes a ‘properties’ parameter that defines which properties will be returned. The properties argument is a vector whose entries are either names of first-class properties or names of functions which take as single required argument a `CommunityCollection` and return a single value. If properties is `NULL`, all first-class properties are included in the returned `data.frame`. Just as with `CPS`, properties can be both ‘first-class’ and computed. `CollectionCPS` is a powerful function that allows you to build up a `data.frame` of predictors and responses. For example, the code fragment below allows us to see how diversity varies with pH.

```
> res <- CollectionCPS(pHWebs, properties=c('pH', 'NumberOfNodes'))
> res
```

	pH	NumberOfNodes
Old Lodge	5.0	23
Afon Hafren	5.3	25
Broadstone	5.5	25
Dargall Lane	5.8	21
Mosedal Beck	5.9	21
Duddon Pike Beck	6.1	35
Allt a'Mharcaidh	6.5	40
Hardknott Gill	7.0	44
Bere Stream	7.5	66
Mill Stream	8.4	87

We can use R’s `lm` function to fit a linear regression model to this data.

```
> model <- lm(NumberOfNodes ~ pH, data=res)
> model
```

Call:  
lm(formula = NumberOfNodes ~ pH, data = res)

Coefficients:  
(Intercept)           pH  
      -85.25           19.68

Let's examine the model's fit to the data.

> summary(model)

Call:  
lm(formula = NumberOfNodes ~ pH, data = res)

Residuals:  
      Min       1Q Median       3Q       Max  
-9.830 -6.556  1.138  5.404  9.878

Coefficients:  
                  Estimate Std. Error t value Pr(>|t|)  
(Intercept) -85.254       14.794  -5.763 0.000423 \*\*\*  
pH            19.675        2.319  8.485 2.85e-05 \*\*\*

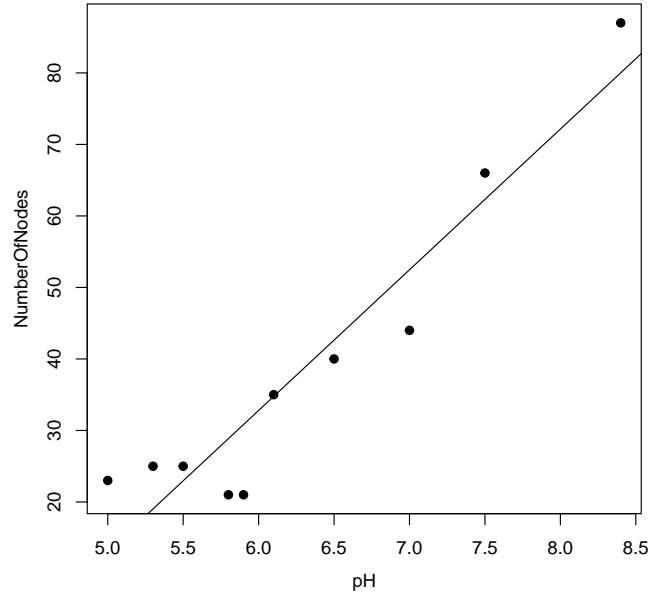
---  
Signif. codes:  0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.391 on 8 degrees of freedom  
Multiple R-squared:  0.9,           Adjusted R-squared:  0.8875  
F-statistic:     72 on 1 and 8 DF,  p-value: 2.852e-05

pH has a significant effect on number of nodes.

Let's plot the data and the model regression line.

```
> with(res, plot(pH, NumberOfNodes, pch=19))
> abline(model)
```



The above figure is similar to (Layer et al., 2010), Fig. 4A (p 281). Cheddar's `pHWebs` dataset contains ten of the twenty food webs analysed by Layer et al. (2010) so the plot is not an exact recreation of the published figure.

The example below uses `CollectionCPS` to assemble a table of four computed properties.

```
> CollectionCPS(pHWebs, c('pH',
                          'NumberOfNodes',
                          'NumberOfTrophicLinks',
                          'DirectedConnectance',
                          'NvMSlope'))
```

	pH	NumberOfNodes	NumberOfTrophicLinks	DirectedConnectance	NvMSlope
Old Lodge	5.0	23	137	0.2589792	-0.6561601
Afon Hafren	5.3	25	135	0.2160000	-0.7078312
Broadstone	5.5	25	178	0.2848000	-0.5853852
Dargall Lane	5.8	21	99	0.2244898	-0.7379515
Mosedal Beck	5.9	21	108	0.2448980	-0.7026522
Duddon Pike Beck	6.1	35	286	0.2334694	-0.5673022
Allt a'Mharcaidh	6.5	40	334	0.2087500	-0.7655290
Hardknott Gill	7.0	44	386	0.1993802	-0.7548597
Bere Stream	7.5	66	943	0.2164830	-0.6501359
Mill Stream	8.4	87	1654	0.2185229	-0.9192528

We can use a named vector to get shorter column titles.

```
> CollectionCPS(pHWebs, c('pH',
                          S='NumberOfNodes',
                          L='NumberOfTrophicLinks',
                          C='DirectedConnectance',
                          Slope='NvMSlope'))
```

	pH	S	L	C	Slope
Old Lodge	5.0	23	137	0.2589792	-0.6561601
Afon Hafren	5.3	25	135	0.2160000	-0.7078312
Broadstone	5.5	25	178	0.2848000	-0.5853852
Dargall Lane	5.8	21	99	0.2244898	-0.7379515
Mosedal Beck	5.9	21	108	0.2448980	-0.7026522
Duddon Pike Beck	6.1	35	286	0.2334694	-0.5673022
Allt a'Mharcaidh	6.5	40	334	0.2087500	-0.7655290
Hardknott Gill	7.0	44	386	0.1993802	-0.7548597
Bere Stream	7.5	66	943	0.2164830	-0.6501359
Mill Stream	8.4	87	1654	0.2185229	-0.9192528

The functions in the above examples each return a single value. Functions are permitted to return more than one value, such as `SumBiomassByClass`, which returns the total biomass in each class; the default class is 'category'. Some pHWebs communities contain nodes (detritus and the like) that do not have a category. These appear in '<unnamed>'. These appear in '<unnamed>'.

```
> CollectionCPS(pHWebs, c('pH',
                          S='NumberOfNodes',
                          L='NumberOfTrophicLinks',
                          C='DirectedConnectance',
                          Slope='NvMSlope',
                          'SumBiomassByClass'))
```

	pH	S	L	C	Slope	<unnamed>	invertebrate	producer
Old Lodge	5.0	23	137	0.2589792	-0.6561601	NA	NA	3.450502e-01
Afon Hafren	5.3	25	135	0.2160000	-0.7078312	NA	321.74544	4.029557e-02
Broadstone	5.5	25	178	0.2848000	-0.5853852	NA	296.95697	2.487247e-03
Dargall Lane	5.8	21	99	0.2244898	-0.7379515	NA	75.99819	1.198280e+00
Mosedal Beck	5.9	21	108	0.2448980	-0.7026522	NA	147.07552	9.363093e-02
Duddon Pike Beck	6.1	35	286	0.2334694	-0.5673022	NA	981.83475	4.037737e-03
Allt a'Mharcaidh	6.5	40	334	0.2087500	-0.7655290	NA	NA	3.933786e+00
Hardknott Gill	7.0	44	386	0.1993802	-0.7548597	NA	522.77646	4.266116e-01
Bere Stream	7.5	66	943	0.2164830	-0.6501359	NA	4991.18212	6.782687e+00
Mill Stream	8.4	87	1654	0.2185229	-0.9192528	NA	NA	4.425616e+02

	vert.ecto
Old Lodge	3500.00000
Afon Hafren	3200.00000
Broadstone	97.50000
Dargall Lane	366666.66667
Mosedal Beck	500.00000
Duddon Pike Beck	300.00000
Allt a'Mharcaidh	412.50000

```

Hardknott Gill      4550.00000
Bere Stream        17.94913
Mill Stream        14200.00000

```

We can use a named vector to prefix column titles of values returned by `SumBiomassByClass`.

```

> CollectionCPS(pHWebs, c('pH',
                          S='NumberOfNodes',
                          L='NumberOfTrophicLinks',
                          C='DirectedConnectance',
                          Slope='NvMSlope',
                          B='SumBiomassByClass'))

```

	pH	S	L	C	Slope	B.<unnamed>	B.invertebrate	B.producer
Old Lodge	5.0	23	137	0.2589792	-0.6561601	NA	NA	3.450502e-01
Afon Hafren	5.3	25	135	0.2160000	-0.7078312	NA	321.74544	4.029557e-02
Broadstone	5.5	25	178	0.2848000	-0.5853852	NA	296.95697	2.487247e-03
Dargall Lane	5.8	21	99	0.2244898	-0.7379515	NA	75.99819	1.198280e+00
Mosedal Beck	5.9	21	108	0.2448980	-0.7026522	NA	147.07552	9.363093e-02
Duddon Pike Beck	6.1	35	286	0.2334694	-0.5673022	NA	981.83475	4.037737e-03
Allt a'Mharcaidh	6.5	40	334	0.2087500	-0.7655290	NA	NA	3.933786e+00
Hardknott Gill	7.0	44	386	0.1993802	-0.7548597	NA	522.77646	4.266116e-01
Bere Stream	7.5	66	943	0.2164830	-0.6501359	NA	4991.18212	6.782687e+00
Mill Stream	8.4	87	1654	0.2185229	-0.9192528	NA	NA	4.425616e+02

```

      B.vert.ecto
Old Lodge      3500.00000
Afon Hafren    3200.00000
Broadstone     97.50000
Dargall Lane   366666.66667
Mosedal Beck   500.00000
Duddon Pike Beck 300.00000
Allt a'Mharcaidh 412.50000
Hardknott Gill 4550.00000
Bere Stream    17.94913
Mill Stream    14200.00000

```

The Old Lodge, Allt a'Mharcaidh and Mill Stream communities each have some invertebrates without *M* and/or *N* either because not enough individuals could be sampled to compute these properties reliably or because no data could be found in the literature. The biomasses for these nodes is NA and the summed biomasses for invertebrates in Old Lodge, Allt a'Mharcaidh and Mill Stream are therefore NA. We can ignore missing values by setting the 'na.rm' parameter.

```

> CollectionCPS(pHWebs, list('pH',
                             S='NumberOfNodes',
                             L='NumberOfTrophicLinks',
                             C='DirectedConnectance',
                             Slope='NvMSlope',
                             B=list('SumBiomassByClass', na.rm=TRUE)))

```

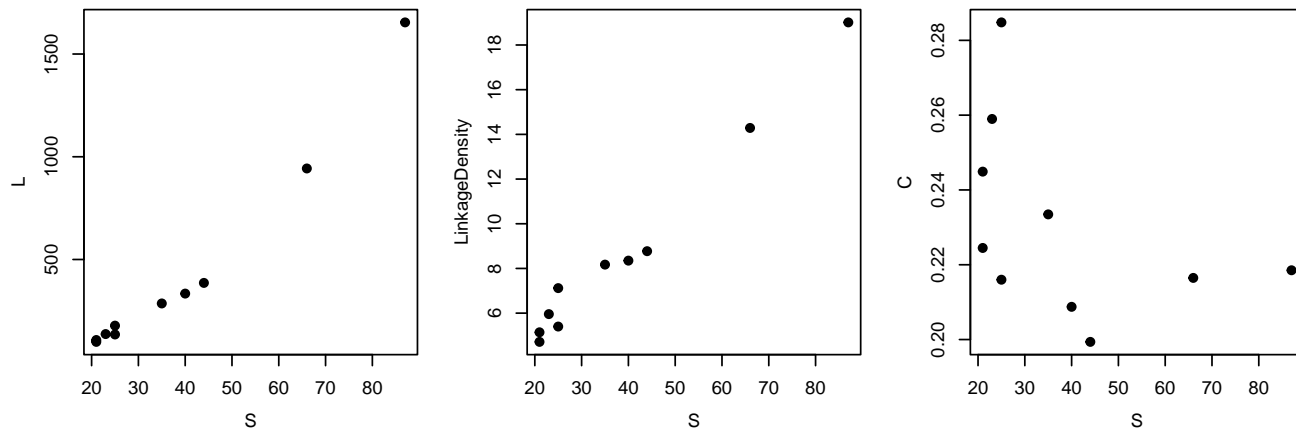




```

> par(mfrow=c(1,3))
> with(properties, plot(S, L, pch=19))
> with(properties, plot(S, LinkageDensity, pch=19))
> with(properties, plot(S, C, pch=19))

```



These plots are similar to those in Riede et al. (2010), Fig. 1 (p 143) and Brown et al. (2011), Fig. 7 (p 891) but using different data.

### 3.4 Node properties

CollectionNPS returns a `data.frame` with a row for every node in every community.

```

> head(CollectionNPS(pHWebs))

```

community	node	M	N	category
1 Old Lodge	CPOM	NA	NA	
2 Old Lodge	FPOM	NA	NA	
3 Old Lodge	Eunotia exigua	1.910441e-12	2067974311	producer
4 Old Lodge	Eunotia rhomboidea	6.820054e-13	210924209	producer
5 Old Lodge	Eunotia vanheurckii	4.290173e-12	862038072	producer
6 Old Lodge	Eunotia incisa	1.992908e-11	220094827	producer

As with CollectionCPS, you can get columns for both first-class and computed properties.

```

> # A subset of first-class properties
> head(CollectionNPS(pHWebs, 'M'))

```

community	node	M
1 Old Lodge	CPOM	NA
2 Old Lodge	FPOM	NA
3 Old Lodge	Eunotia exigua	1.910441e-12
4 Old Lodge	Eunotia rhomboidea	6.820054e-13
5 Old Lodge	Eunotia vanheurckii	4.290173e-12
6 Old Lodge	Eunotia incisa	1.992908e-11

```
> # Several properties
> head(CollectionNPS(pHWebs, c('M', 'N', 'Biomass', 'Degree', 'IsBasalNode')))
```

community	node	M	N	Biomass	Degree	IsBasalNode
1 Old Lodge	CPOM	NA	NA	NA	4	TRUE
2 Old Lodge	FPOM	NA	NA	NA	8	TRUE
3 Old Lodge	Eunotia exigua	1.910441e-12	2067974311	0.0039507435	9	TRUE
4 Old Lodge	Eunotia rhomboidea	6.820054e-13	210924209	0.0001438514	9	TRUE
5 Old Lodge	Eunotia vanheurckii	4.290173e-12	862038072	0.0036982924	9	TRUE
6 Old Lodge	Eunotia incisa	1.992908e-11	220094827	0.0043862864	9	TRUE

```
> # Named properties
> head(CollectionNPS(pHWebs, c('M', 'N', B='Biomass', 'Degree', Basal='IsBasalNode')))
```

community	node	M	N	B	Degree	Basal
1 Old Lodge	CPOM	NA	NA	NA	4	TRUE
2 Old Lodge	FPOM	NA	NA	NA	8	TRUE
3 Old Lodge	Eunotia exigua	1.910441e-12	2067974311	0.0039507435	9	TRUE
4 Old Lodge	Eunotia rhomboidea	6.820054e-13	210924209	0.0001438514	9	TRUE
5 Old Lodge	Eunotia vanheurckii	4.290173e-12	862038072	0.0036982924	9	TRUE
6 Old Lodge	Eunotia incisa	1.992908e-11	220094827	0.0043862864	9	TRUE

### 3.5 Trophic link properties

CollectionTLPS returns a data.frame containing a row for every trophic link in every community:

```
> head(CollectionTLPS(pHWebs))
```

community	resource	consumer
1 Old Lodge	Chironomidae undet.	Siphonoperla torrentium
2 Old Lodge	Leuctra nigra	Siphonoperla torrentium
3 Old Lodge	Nemoura cinerea	Siphonoperla torrentium
4 Old Lodge	Simuliidae gra	Siphonoperla torrentium
5 Old Lodge	Simuliidae grb	Siphonoperla torrentium
6 Old Lodge	Simuliidae grc	Siphonoperla torrentium

Community names and resource and consumer *M*:

```
> head(CollectionTLPS(pHWebs, 'M'))
```

community	resource	consumer	resource.M	consumer.M
1 Old Lodge	Chironomidae undet.	Siphonoperla torrentium	0.07809028	1.03837
2 Old Lodge	Leuctra nigra	Siphonoperla torrentium	0.94124878	1.03837
3 Old Lodge	Nemoura cinerea	Siphonoperla torrentium	0.35536589	1.03837
4 Old Lodge	Simuliidae gra	Siphonoperla torrentium	0.17317064	1.03837
5 Old Lodge	Simuliidae grb	Siphonoperla torrentium	0.18357370	1.03837
6 Old Lodge	Simuliidae grc	Siphonoperla torrentium	0.22403767	1.03837

Several properties:

```
> head(CollectionTLPS(pHWebs, c('M', 'N', 'Biomass', 'Degree', 'IsBasalNode')))
```

	community	resource	consumer	resource.M	resource.N
1	Old Lodge	Chironomidae undet.	Siphonoperla torrentium	0.07809028	64.0
2	Old Lodge	Leuctra nigra	Siphonoperla torrentium	0.94124878	214.4
3	Old Lodge	Nemoura cinerea	Siphonoperla torrentium	0.35536589	342.4
4	Old Lodge	Simuliidae gra	Siphonoperla torrentium	0.17317064	6.4
5	Old Lodge	Simuliidae grb	Siphonoperla torrentium	0.18357370	83.2
6	Old Lodge	Simuliidae grc	Siphonoperla torrentium	0.22403767	3.2
	resource.Biomass	resource.Degree	resource.IsBasalNode	consumer.M	consumer.N
1	4.9977782	16	FALSE	1.03837	16
2	201.8037377	16	FALSE	1.03837	16
3	121.6772793	16	FALSE	1.03837	16
4	1.1082921	15	FALSE	1.03837	16
5	15.2733322	15	FALSE	1.03837	16
6	0.7169205	15	FALSE	1.03837	16
	consumer.Biomass	consumer.Degree	consumer.IsBasalNode		
1	16.61392	11	FALSE		
2	16.61392	11	FALSE		
3	16.61392	11	FALSE		
4	16.61392	11	FALSE		
5	16.61392	11	FALSE		
6	16.61392	11	FALSE		

Several properties with shorter column names:

```
> head(CollectionTLPS(pHWebs, c('M','N', B='Biomass', D='Degree', Basal='IsBasalNode')))
```

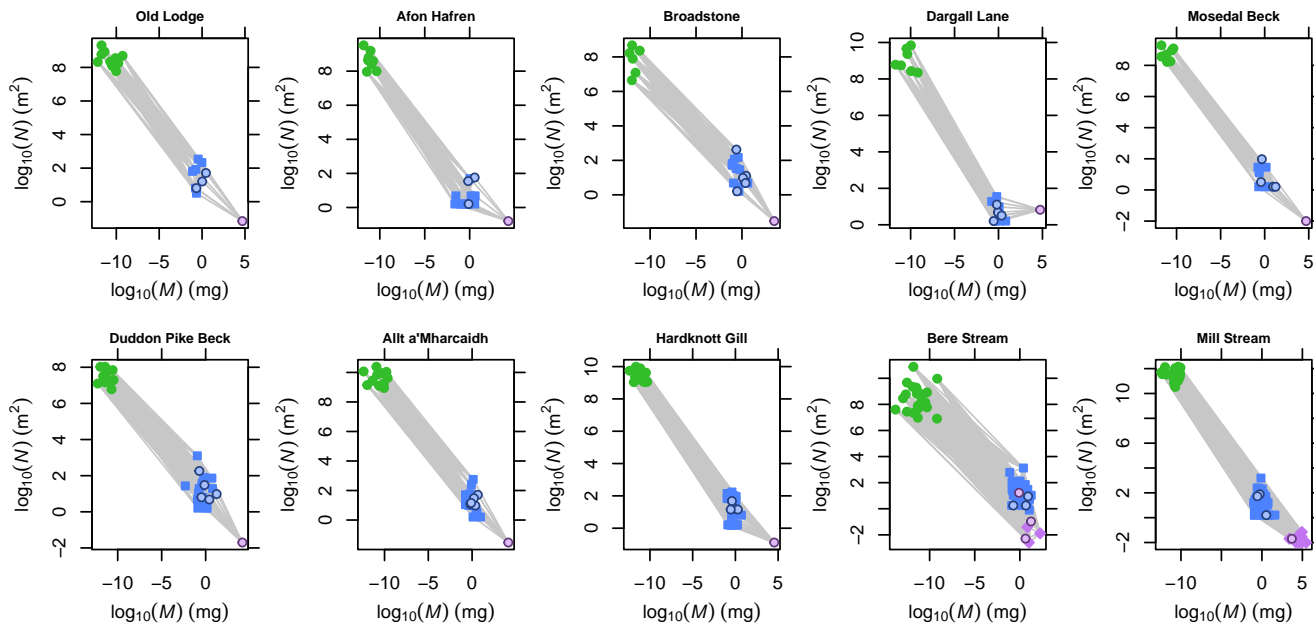
	community	resource	consumer	resource.M	resource.N	resource.B	
1	Old Lodge	Chironomidae undet.	Siphonoperla torrentium	0.07809028	64.0	4.9977782	
2	Old Lodge	Leuctra nigra	Siphonoperla torrentium	0.94124878	214.4	201.8037377	
3	Old Lodge	Nemoura cinerea	Siphonoperla torrentium	0.35536589	342.4	121.6772793	
4	Old Lodge	Simuliidae gra	Siphonoperla torrentium	0.17317064	6.4	1.1082921	
5	Old Lodge	Simuliidae grb	Siphonoperla torrentium	0.18357370	83.2	15.2733322	
6	Old Lodge	Simuliidae grc	Siphonoperla torrentium	0.22403767	3.2	0.7169205	
	resource.D	resource.Basal	consumer.M	consumer.N	consumer.B	consumer.D	consumer.Basal
1	16	FALSE	1.03837	16	16.61392	11	FALSE
2	16	FALSE	1.03837	16	16.61392	11	FALSE
3	16	FALSE	1.03837	16	16.61392	11	FALSE
4	15	FALSE	1.03837	16	16.61392	11	FALSE
5	15	FALSE	1.03837	16	16.61392	11	FALSE
6	15	FALSE	1.03837	16	16.61392	11	FALSE

## 4 Plotting

### 4.1 Plot-per-community

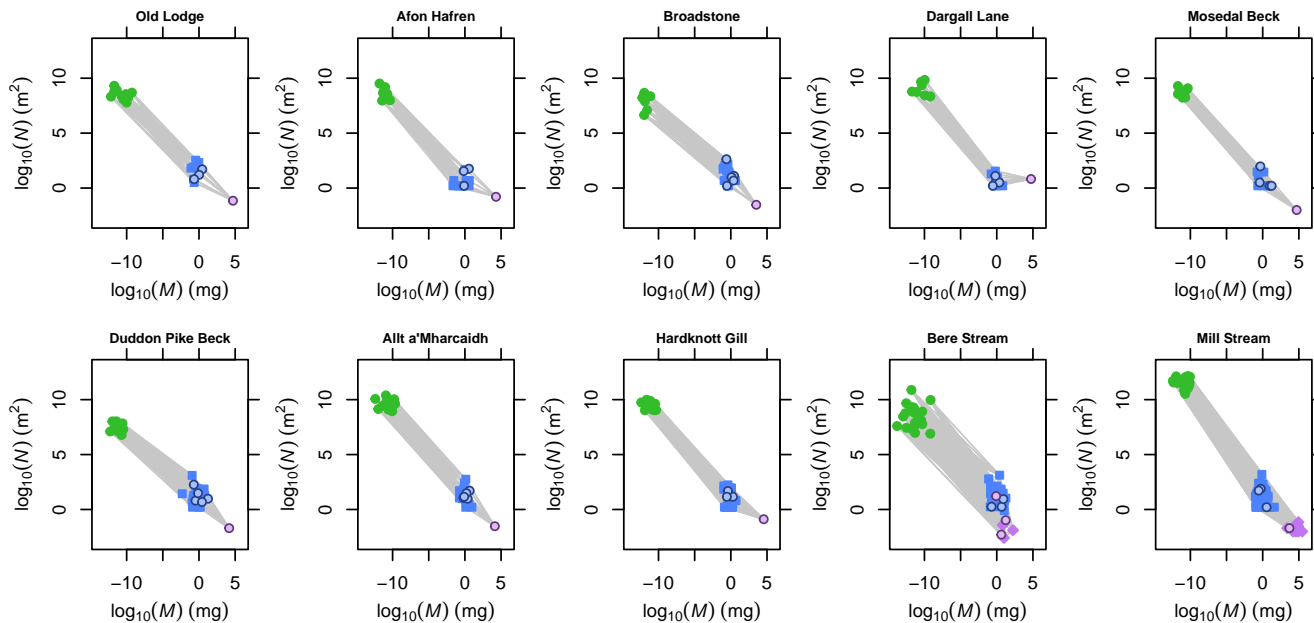
You can use R's plot function to 'eyeball' webs in a collection.

```
> plot(pHWebs)
```



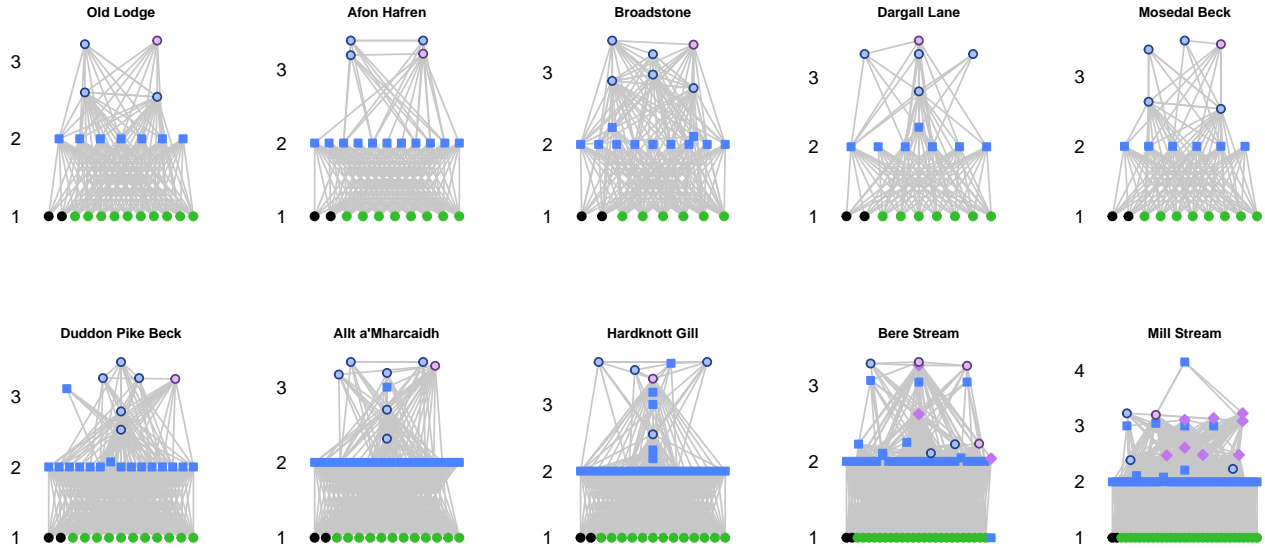
You can use R's plot parameters 'xlim' and 'ylim' to set limits for the x and y axes.

```
> plot(pHWebs, xlim=c(-14,6), ylim=c(-3,13))
```



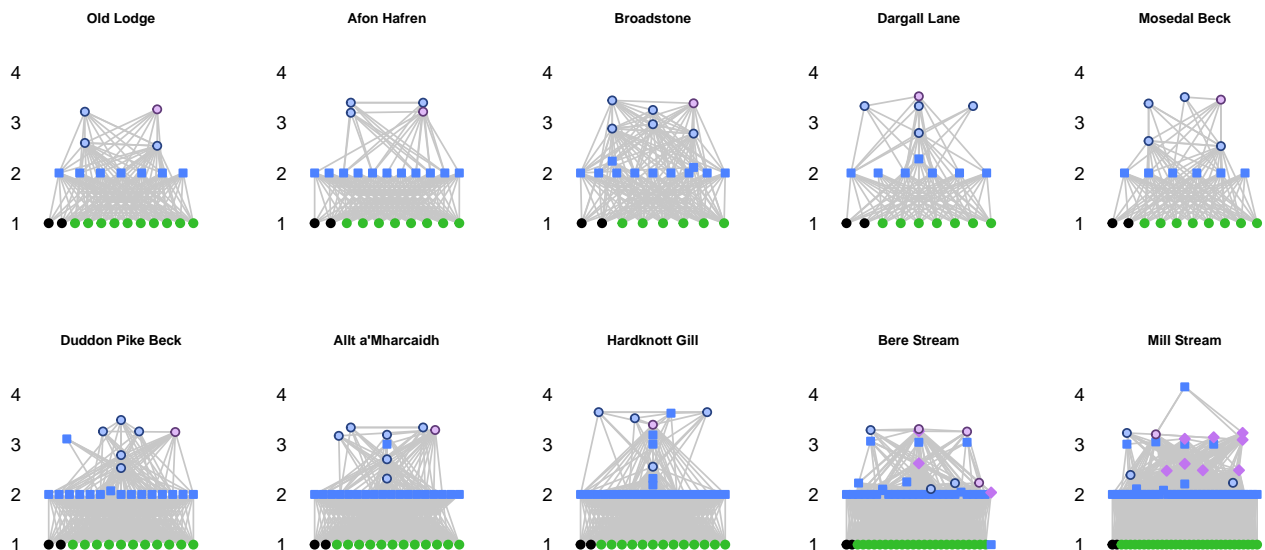
Cheddar examines the properties of the communities in the collection in order to decide which `Community`-level plot function to use. You can change this behaviour using the `'plot.fn'` parameter. The `PlotWebByLevel` allows the webs to be viewed by trophic level.

```
> plot(pHWebs, plot.fn=PlotWebByLevel)
```



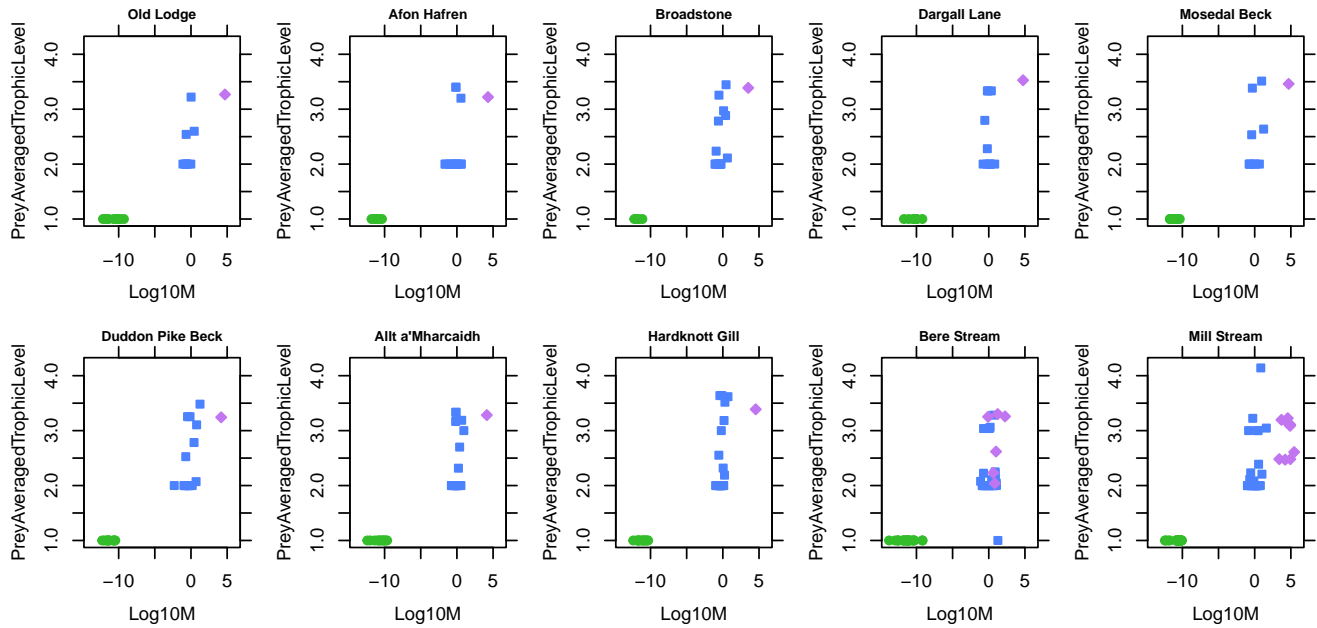
As in the previous example, the y axis limits can be made consistent.

```
> plot(pHWebs, plot.fn=PlotWebByLevel, ylim=c(1, 4.5))
```



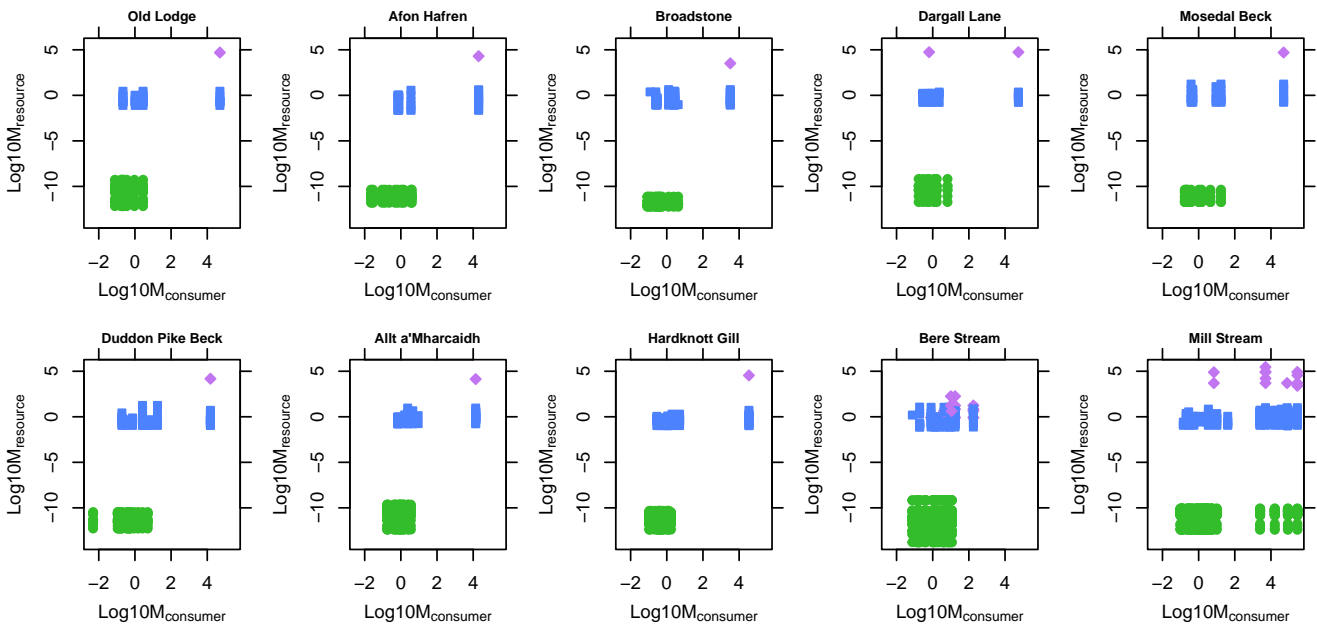
We can use the general-purpose function `PlotNPS` to plot any node properties that we like and all of the power of `PlotNPS` is available. The example below plots trophic level as a function of  $\log_{10}$ -transformed body mass. Each plot has the same axis limits. We have turned off plotting of the food web and highlighting of cannibals.

```
> plot(pHWebs, plot.fn=PlotNPS, X='Log10M', Y='PreyAveragedTrophicLevel',
      show.web=FALSE, highlight.nodes=NULL, xlim=c(-14,6), ylim=c(1,4.2))
```



We can also use `PlotTLPS`, as shown below.

```
> plot(pHWebs, plot.fn=PlotTLPS, X='consumer.Log10M',
      Y='resource.Log10M', xlim=c(-2.5, 5.5), ylim=c(-13.8, 5.5))
```



## 5 Modifying communities

The `CollectionApply` function allows communities within collections to be modified. For example, with certain analyses it can be desirable to remove isolated nodes.

```
> # Bere Stream has some isolated nodes
> CollectionCPS(pHWebs, 'FractionIsolatedNodes')
```

	FractionIsolatedNodes
Old Lodge	0.00000000
Afon Hafren	0.00000000
Broadstone	0.00000000
Dargall Lane	0.00000000
Mosedal Beck	0.00000000
Duddon Pike Beck	0.00000000
Allt a'Mharcaidh	0.00000000
Hardknott Gill	0.00000000
Bere Stream	0.01515152
Mill Stream	0.00000000

```
> pHWebs.no.iso <- CollectionApply(pHWebs, RemoveIsolatedNodes)
> CollectionCPS(pHWebs.no.iso, 'FractionIsolatedNodes') # All 0
```

	FractionIsolatedNodes
Old Lodge	0
Afon Hafren	0
Broadstone	0
Dargall Lane	0
Mosedal Beck	0
Duddon Pike Beck	0
Allt a'Mharcaidh	0
Hardknott Gill	0
Bere Stream (isolated nodes removed)	0
Mill Stream	0

The `CollectionApply` function can be used with any function that modifies communities, such as `RemoveCannibalisticLinks`.

```
> # The number of cannibals in each community
> sapply(pHWebs, function(community) length(Cannibals(community)))
```

Old Lodge	Afon Hafren	Broadstone	Dargall Lane	Mosedal Beck
4	4	6	5	5
Duddon Pike Beck	Allt a'Mharcaidh	Hardknott Gill	Bere Stream	Mill Stream
6	7	5	6	4

```
> pHWebs.no.can <- CollectionApply(pHWebs, RemoveCannibalisticLinks)
> sapply(pHWebs.no.can, function(community) length(Cannibals(community)))
```



```

    Old Lodge (cannibalistic links removed)
                                0
    Afon Hafren (cannibalistic links removed)
                                0
    Broadstone (cannibalistic links removed)
                                0
    Dargall Lane (cannibalistic links removed)
                                0
    Mosedal Beck (cannibalistic links removed)
                                0
    Duddon Pike Beck (cannibalistic links removed)
                                0
    Allt a'Mharcaidh (cannibalistic links removed)
                                0
    Hardknott Gill (cannibalistic links removed)
                                0
    Bere Stream (cannibalistic links removed)
                                0
    Mill Stream (cannibalistic links removed)
                                0

```

The function to be applied to each community can also take additional parameters. The following example reorders each community's nodes by body mass.

```
> head(CollectionNPS(pHWebs))
```

	community	node	M	N	category
1	Old Lodge	CPOM	NA	NA	
2	Old Lodge	FPOM	NA	NA	
3	Old Lodge	Eunotia exigua	1.910441e-12	2067974311	producer
4	Old Lodge	Eunotia rhomboidea	6.820054e-13	210924209	producer
5	Old Lodge	Eunotia vanheurckii	4.290173e-12	862038072	producer
6	Old Lodge	Eunotia incisa	1.992908e-11	220094827	producer

```
> pHWebs.by.M <- CollectionApply(pHWebs, OrderCommunity, 'M')
```

```
> head(CollectionNPS(pHWebs.by.M))
```

	community	node	M	N	category
1	Old Lodge (reordered)	Eunotia rhomboidea	6.820054e-13	210924209	producer
2	Old Lodge (reordered)	Eunotia exigua	1.910441e-12	2067974311	producer
3	Old Lodge (reordered)	Brachysira vitrea	1.910441e-12	600675465	producer
4	Old Lodge (reordered)	Eunotia vanheurckii	4.290173e-12	862038072	producer
5	Old Lodge (reordered)	Eunotia incisa	1.992908e-11	220094827	producer
6	Old Lodge (reordered)	Brachysira brebissonii	3.426856e-11	119218031	producer

We can put the nodes lacking *M* first.

```
> pHWebs.by.M <- CollectionApply(pHWebs, OrderCommunity, 'M', na.last=FALSE)
```

```
> head(CollectionNPS(pHWebs.by.M))
```

	community	node	M	N	category
1	Old Lodge (reordered)	CPOM	NA	NA	
2	Old Lodge (reordered)	FPOM	NA	NA	
3	Old Lodge (reordered)	Leuctra sp.	NA	NA	invertebrate
4	Old Lodge (reordered)	Eunotia rhomboidea	6.820054e-13	210924209	producer
5	Old Lodge (reordered)	Eunotia exigua	1.910441e-12	2067974311	producer
6	Old Lodge (reordered)	Brachysira vitrea	1.910441e-12	600675465	producer

## 6 Ordering collections

OrderCollection allows you to order collections by whatever properties you please. To order the webs by decreasing pH:

```
> pHWebs.decreasing.pH <- OrderCollection(pHWebs, 'pH', decreasing=TRUE)
> CollectionCPS(pHWebs.decreasing.pH)
```

	title	M.units	N.units	code	pH	lat	long
Mill Stream	Mill Stream	mg	m <sup>2</sup>	MIL	8.4	50.68	-2.180
Bere Stream	Bere Stream	mg	m <sup>2</sup>	BER	7.5	50.73	-2.210
Hardknott Gill	Hardknott Gill	mg	m <sup>2</sup>	DUD2	7.0	54.40	-3.170
Allt a'Mharcaidh	Allt a'Mharcaidh	mg	m <sup>2</sup>	MHA	6.5	57.12	-3.850
Duddon Pike Beck	Duddon Pike Beck	mg	m <sup>2</sup>	DUD1	6.1	54.41	-3.170
Mosedal Beck	Mosedal Beck	mg	m <sup>2</sup>	DUD3	5.9	54.41	-3.140
Dargall Lane	Dargall Lane	mg	m <sup>2</sup>	DAR	5.8	55.08	-4.430
Broadstone	Broadstone	mg	m <sup>2</sup>	BRO	5.5	51.08	0.053
Afon Hafren	Afon Hafren	mg	m <sup>2</sup>	HAF	5.3	52.47	-3.700
Old Lodge	Old Lodge	mg	m <sup>2</sup>	OLD	5.0	51.04	0.080

To order alphabetically by community name.

```
> pHWebs.name <- OrderCollection(pHWebs, 'title')
> CollectionCPS(pHWebs.name)
```

	title	M.units	N.units	code	pH	lat	long
Afon Hafren	Afon Hafren	mg	m <sup>2</sup>	HAF	5.3	52.47	-3.700
Allt a'Mharcaidh	Allt a'Mharcaidh	mg	m <sup>2</sup>	MHA	6.5	57.12	-3.850
Bere Stream	Bere Stream	mg	m <sup>2</sup>	BER	7.5	50.73	-2.210
Broadstone	Broadstone	mg	m <sup>2</sup>	BRO	5.5	51.08	0.053
Dargall Lane	Dargall Lane	mg	m <sup>2</sup>	DAR	5.8	55.08	-4.430
Duddon Pike Beck	Duddon Pike Beck	mg	m <sup>2</sup>	DUD1	6.1	54.41	-3.170
Hardknott Gill	Hardknott Gill	mg	m <sup>2</sup>	DUD2	7.0	54.40	-3.170
Mill Stream	Mill Stream	mg	m <sup>2</sup>	MIL	8.4	50.68	-2.180
Mosedal Beck	Mosedal Beck	mg	m <sup>2</sup>	DUD3	5.9	54.41	-3.140
Old Lodge	Old Lodge	mg	m <sup>2</sup>	OLD	5.0	51.04	0.080

You can sort on computed properties, such as the number of nodes.

```
> pHWebs.n.nodes <- OrderCollection(pHWebs, 'NumberOfNodes')
> CollectionCPS(pHWebs.n.nodes, c('pH', 'lat', 'NumberOfNodes'))
```

	pH	lat	NumberOfNodes
Dargall Lane	5.8	55.08	21
Mosedal Beck	5.9	54.41	21
Old Lodge	5.0	51.04	23
Afon Hafren	5.3	52.47	25
Broadstone	5.5	51.08	25
Duddon Pike Beck	6.1	54.41	35
Allt a'Mharcaidh	6.5	57.12	40

Hardknott Gill	7.0	54.40	44
Bere Stream	7.5	50.73	66
Mill Stream	8.4	50.68	87

Two communities have 21 nodes and two have 25. We can sort on more than one property to break ties. This example sorts by number of nodes and the latitude within number of nodes.

```
> pHWebs.n.nodes.and.lat <- OrderCollection(pHWebs, 'NumberOfNodes', 'lat')
> CollectionCPS(pHWebs.n.nodes.and.lat, c('pH', 'lat', 'NumberOfNodes'))
```

	pH	lat	NumberOfNodes
Mosedal Beck	5.9	54.41	21
Dargall Lane	5.8	55.08	21
Old Lodge	5.0	51.04	23
Broadstone	5.5	51.08	25
Afon Hafren	5.3	52.47	25
Duddon Pike Beck	6.1	54.41	35
Allt a'Mharcaidh	6.5	57.12	40
Hardknott Gill	7.0	54.40	44
Bere Stream	7.5	50.73	66
Mill Stream	8.4	50.68	87

## 7 Aggregating communities

`AggregateCommunities` aggregates the communities within a collection in to a new single community object. The way that node, trophic link and community properties are aggregated are shown here using the `Millstream` data set (?Ledger et al., 2011). The ‘c4’ community was a control and the ‘d4’ community was exposed to a drought treatment.

```
> data(Millstream)
> Millstream
```

A collection of 2 communities

```
> names(Millstream)
```

```
[1] "c4" "d4"
```

The herbivorous insect *Synorthocladius sp.* appears in both communities but with a different mean  $M$  and  $N$ .

```
> nps <- CollectionNPS(Millstream)
> nps['Synorthocladius sp.'==nps$node,c('community','M','N')]
```

	community	M	N
56	c4	0.02099907	32.92305
114	d4	0.03868121	206.58558

Now let’s perform the aggregation of these two communities, weighting by  $N$ :

```
> aggregation1 <- AggregateCommunities(Millstream, weight.by='N')
> # Satisfy ourselves that each node has been included in the aggregated community
> all(sort(unique(nps$node))==sort(NPS(aggregation1)$node))
```

```
[1] TRUE
```

Now let’s examine how ‘M’ and ‘N’ have been computed for *Synorthocladius sp.*:

```
> NPS(aggregation1)['Synorthocladius sp.',c('M','N')]
```

	M	N
Synorthocladius sp.	0.0362506	119.7543

These values were computed from the values in the collection as follows:

```
> # Arithmetic mean of N
> mean(nps['Synorthocladius sp.'==nps$node,'N'])
```

```
[1] 119.7543
```

```
> # N-weighted mean of M
> weighted.mean(nps['Synorthocladius sp.'==nps$node,'M'],
               nps['Synorthocladius sp.'==nps$node,'N'])
```

```
[1] 0.0362506
```

Now let's see what happens when we perform the aggregation of these two communities without any weighting:

```
> aggregation2 <- AggregateCommunities(Millstream, weight.by=NULL)
> NPS(aggregation2)['Synorthocladius sp.',c('M','N')]
```

```
                M          N
Synorthocladius sp. 0.02984014 119.7543
```

```
> # Arithmetic mean of M
> mean(nps['Synorthocladius sp.'==nps$node,'M'])
```

```
[1] 0.02984014
```

```
> # Arithmetic mean of N
> mean(nps['Synorthocladius sp.'==nps$node,'N'])
```

```
[1] 119.7543
```

`AggregateCommunities` combines character and logical node properties by joining unique values with a `'.'`. `AggregateCommunities` aggregates trophic links by taking the union of links across all communities. There are twelve trophic links in to and out of *Synorthocladius sp.* in 'c4' and 'd4'.

```
> tlps <- CollectionTLPS(Millstream)
> tlps['Synorthocladius sp.'==tlps$resource |
      'Synorthocladius sp.'==tlps$consumer,]
```

	community	resource	consumer
283	c4	Synorthocladius sp.	Polycentropus flavomaculatus
355	c4	Amorphous detritus (FPOM)	Synorthocladius sp.
356	c4	Plant fragments (CPOM)	Synorthocladius sp.
357	c4	Navicula gregaria	Synorthocladius sp.
358	c4	Navicula tripunctata	Synorthocladius sp.
359	c4	Gomphonema olivaceum	Synorthocladius sp.
360	c4	Cocconeis placentula	Synorthocladius sp.
361	c4	Rhoicosphenia abbreviata	Synorthocladius sp.
362	c4	Gongrosira incrustans	Synorthocladius sp.
617	d4	Amorphous detritus (FPOM)	Synorthocladius sp.
618	d4	Plant fragments (CPOM)	Synorthocladius sp.
619	d4	Gongrosira incrustans	Synorthocladius sp.

The union of these twelve trophic links gives nine unique links:

```
> TrophicLinksForNodes(aggregation1, 'Synorthocladius sp.')
```

	resource	consumer
1	Synorthocladius sp.	Polycentropus flavomaculatus
2	Amorphous detritus (FPOM)	Synorthocladius sp.
3	Plant fragments (CPOM)	Synorthocladius sp.

4	Navicula gregaria	Synorthocladus sp.
5	Navicula tripunctata	Synorthocladus sp.
6	Gomphonema olivaceum	Synorthocladus sp.
7	Cocconeis placentula	Synorthocladus sp.
8	Rhoicosphenia abbreviata	Synorthocladus sp.
9	Gongrosira incrustans	Synorthocladus sp.

Community properties are aggregated by computing the arithmetic mean of numeric values and by joining unique character and logical together with a ',':

```
> CollectionCPS(Millstream)
```

	title	M.units	N.units	treatment	block
c4	c4	mg	m <sup>-2</sup>	control	4
d4	d4	mg	m <sup>-2</sup>	disturbed	4

```
> data.frame(CPS(aggregation1))
```

	title	M.units	N.units	treatment	block
1	Aggregation of c4,d4	mg	m <sup>-2</sup>	control,disturbed	4

AggregateCommunitiesBy aggregates by a single property, either first-class or computed, of the contained communities. Each food web in the pHWebs dataset has a different pH, so aggregating by pH would result in a collection of the same ten communities. The Duddon Pike Beck and Mosedal Beck communities share the same latitude and have pH values of 6.1 and 5.9 respectively.

```
> CollectionCPS(pHWebs[c('Duddon Pike Beck', 'Mosedal Beck')])
```

	title	M.units	N.units	code	pH	lat	long
Duddon Pike Beck	Duddon Pike Beck	mg	m <sup>2</sup>	DUD1	6.1	54.41	-3.17
Mosedal Beck	Mosedal Beck	mg	m <sup>2</sup>	DUD3	5.9	54.41	-3.14

Aggregating by the 'lat' property therefore results in a new collection of nine communities.

```
> CollectionCPS(AggregateCommunitiesBy(pHWebs, 'lat'))
```

	title	M.units	N.units	code	pH	lat	long
Aggregation of Old Lodge	Aggregation of Old Lodge						
Aggregation of Afon Hafren	Aggregation of Afon Hafren						
Aggregation of Broadstone	Aggregation of Broadstone						
Aggregation of Dargall Lane	Aggregation of Dargall Lane						
Aggregation of Mosedal Beck,Duddon Pike Beck	Aggregation of Mosedal Beck,Duddon Pike Beck						
Aggregation of Allt a'Mharcaidh	Aggregation of Allt a'Mharcaidh						
Aggregation of Hardknott Gill	Aggregation of Hardknott Gill						
Aggregation of Bere Stream	Aggregation of Bere Stream						
Aggregation of Mill Stream	Aggregation of Mill Stream						
Aggregation of Old Lodge		mg	m <sup>2</sup>	OLD	5.0	51.04	0.080
Aggregation of Afon Hafren		mg	m <sup>2</sup>	HAF	5.3	52.47	-3.700
Aggregation of Broadstone		mg	m <sup>2</sup>	BRO	5.5	51.08	0.053

Aggregation of Dargall Lane	mg	m <sup>2</sup>	DAR	5.8	55.08	-4.430
Aggregation of Mosedal Beck,Duddon Pike Beck	mg	m <sup>2</sup>	DUD3,DUD1	6.0	54.41	-3.155
Aggregation of Allt a'Mharcaidh	mg	m <sup>2</sup>	MHA	6.5	57.12	-3.850
Aggregation of Hardknott Gill	mg	m <sup>2</sup>	DUD2	7.0	54.40	-3.170
Aggregation of Bere Stream	mg	m <sup>2</sup>	BER	7.5	50.73	-2.210
Aggregation of Mill Stream	mg	m <sup>2</sup>	MIL	8.4	50.68	-2.180

The aggregation of Duddon Pike Beck and Mosedal Beck has a pH of 6: the arithmetic mean of the two pH values of the two communities.



## 8 ‘Global’ node IDs

This section describes how to assign a unique ID number to every species in a `CommunityCollection`. This is a common requirement for studies of multiple communities.

### 8.1 Create IDs

This code fragment creates a mapping from species names to global IDs. The IDs are assigned starting with producers, then invertebrates, then fish, sorted alphabetically within each category.

```
> data(TL84, TL86)
> TL <- CommunityCollection(list(TL84, TL86))
> # TL.aggregated is a new Community object containing every species in the TL
> all.TL <- AggregateCommunities(TL)
> # Generate a factor of categories
> nps <- NPS(all.TL, c('node', 'category'))
> categories <- factor(nps$category, levels=c('producer', 'invertebrate',
                                             'vert.ecto'))

> # Order all.TL by categories
> all.TL <- OrderCommunity(all.TL, new.order=order(categories, nps$node))
> # Create the mapping from node name to ID
> map <- 1:NumberOfNodes(all.TL)
> names(map) <- unname(NP(all.TL, 'node'))
```

### 8.2 Table of properties

This code fragment creates a table showing species’ names, categories and IDs.

```
> data.frame(ID=1:NumberOfNodes(all.TL),
             NPS(all.TL, c(Species='node', Category='category',
                          'M', 'N', TL='PreyAveragedTrophicLevel')),
             row.names=NULL)
```

	ID	Species	Category	M	N	TL
1	1	Anabaena circinalis	producer	1.910000e-13	3.000e+06	1.000000
2	2	Ankyra judayi	producer	1.530000e-13	6.500e+06	1.000000
3	3	Arthrodesmus sp.	producer	1.520000e-12	2.450e+07	1.000000
4	4	Asterionella formosa	producer	1.120000e-12	2.500e+06	1.000000
5	5	Chromulina sp.	producer	3.030000e-14	1.790e+08	1.000000
6	6	Chroococcus dispersus	producer	2.390000e-13	1.250e+07	1.000000
7	7	Chroococcus limneticus	producer	1.310000e-12	8.000e+06	1.000000
8	8	Chrysosphaerella longispina	producer	6.736000e-10	2.500e+06	1.000000
9	9	Closteriopsis longissimus	producer	2.366139e-13	5.050e+07	1.000000
10	10	Cosmarium sp.	producer	3.710000e-12	5.000e+05	1.000000
11	11	Cryptomonas sp. 1	producer	2.308969e-13	4.850e+07	1.000000
12	12	Cryptomonas sp. 2	producer	1.510000e-12	1.400e+07	1.000000
13	13	Cryptomonas sp. 3	producer	6.720000e-13	9.000e+06	1.000000
14	14	Cryptomonas sp. 4	producer	1.640000e-12	1.400e+07	1.000000
15	15	Dactylococcopsis fascicularis	producer	1.320000e-13	2.350e+07	1.000000
16	16	Diceras sp.	producer	1.530000e-13	7.500e+06	1.000000

17	17	<i>Dictyosphaerium pulchellum</i>	producer	4.998571e-13	7.000e+06	1.000000
18	18	<i>Dinobryon bavaricum</i>	producer	3.040000e-12	1.900e+07	1.000000
19	19	<i>Dinobryon cylindricum</i>	producer	4.355286e-12	3.500e+07	1.000000
20	20	<i>Dinobryon sertularia</i>	producer	1.074000e-11	2.000e+06	1.000000
21	21	<i>Dinobryon sociale</i>	producer	6.410000e-13	1.400e+07	1.000000
22	22	<i>Glenodinium pulvisculus</i>	producer	5.200000e-12	4.000e+06	1.000000
23	23	<i>Glenodinium quadridens</i>	producer	7.540000e-12	3.350e+07	1.000000
24	24	<i>Gloeocystis</i> sp.	producer	9.460000e-11	2.500e+06	1.000000
25	25	<i>Mallomonas</i> sp. 1	producer	1.030000e-12	9.500e+06	1.000000
26	26	<i>Mallomonas</i> sp. 2	producer	1.410000e-12	1.135e+07	1.000000
27	27	<i>Mallomonas</i> -spiny sp. 1	producer	2.220000e-12	1.050e+07	1.000000
28	28	<i>Mallomonas</i> -spiny sp. 2	producer	2.220000e-12	1.300e+07	1.000000
29	29	<i>Microcystis aeruginosa</i>	producer	1.620000e-11	7.500e+06	1.000000
30	30	<i>Nostoc</i> sp.	producer	7.970000e-13	1.000e+06	1.000000
31	31	<i>Oocystis</i> sp. 1	producer	3.860000e-12	1.200e+07	1.000000
32	32	<i>Oocystis</i> sp. 2	producer	6.320000e-12	1.500e+06	1.000000
33	33	<i>Oscillatoria</i> sp.	producer	1.610000e-12	3.000e+06	1.000000
34	34	<i>Peridinium cinctum</i>	producer	4.060000e-11	4.000e+06	1.000000
35	35	<i>Peridinium limbatum</i>	producer	6.460000e-11	9.000e+06	1.000000
36	36	<i>Peridinium pulsillum</i>	producer	1.580000e-12	6.200e+07	1.000000
37	37	<i>Peridinium wisconsinense</i>	producer	3.560000e-11	7.000e+06	1.000000
38	38	<i>Quadrigula lacustris</i>	producer	7.130000e-12	5.150e+07	1.000000
39	39	<i>Quadrigula</i> sp. 2	producer	9.480000e-13	5.500e+06	1.000000
40	40	<i>Rhizosolenia</i> sp.	producer	6.860000e-13	2.800e+07	1.000000
41	41	<i>Schroederia setigera</i>	producer	6.370000e-13	4.450e+07	1.000000
42	42	<i>Selenastrum minutum</i>	producer	2.720000e-13	1.055e+08	1.000000
43	43	<i>Sphaerocystis schroeteri</i>	producer	1.080000e-11	1.000e+06	1.000000
44	44	<i>Spinocosmarium</i> sp.	producer	3.710000e-12	1.000e+06	1.000000
45	45	<i>Staurastrum</i> sp.	producer	4.300000e-12	4.500e+06	1.000000
46	46	<i>Synedra</i> sp.	producer	3.087467e-11	1.500e+06	1.000000
47	47	<i>Synura</i> sp.	producer	5.070000e-12	5.000e+05	1.000000
48	48	<i>Trachelomonas</i> sp.	producer	1.750000e-13	1.110e+08	1.000000
49	49	Unclassified flagellates	producer	3.460000e-13	9.400e+08	1.000000
50	50	Unclassified microflagellates	producer	1.020000e-13	6.300e+07	1.000000
51	51	<i>Ascomorpha eucadis</i>	invertebrate	2.968966e-10	1.740e+04	2.000000
52	52	<i>Bosmina longirostris</i>	invertebrate	1.550000e-09	7.750e+04	2.000000
53	53	<i>Chaoborus punctipennis</i>	invertebrate	2.550000e-07	1.200e+04	3.142379
54	54	<i>Conochiloides dossuarius</i>	invertebrate	1.600000e-10	1.184e+05	2.000000
55	55	<i>Conochilus</i> (colonial)	invertebrate	1.460000e-08	2.100e+03	2.000000
56	56	<i>Conochilus</i> (solitary)	invertebrate	3.500000e-11	1.800e+04	2.000000
57	57	<i>Cyclops varians rubellus</i>	invertebrate	2.134118e-08	5.100e+03	3.166667
58	58	<i>Daphnia pulex</i>	invertebrate	4.697241e-08	8.700e+03	2.071429
59	59	<i>Daphnia rosea</i>	invertebrate	1.360000e-08	1.200e+03	2.130435
60	60	<i>Diaphanosoma leuchtenbergianum</i>	invertebrate	2.240000e-09	7.200e+03	2.000000
61	61	<i>Diaptomus oregonensis</i>	invertebrate	1.440000e-08	3.000e+02	2.000000
62	62	<i>Filinia longispina</i>	invertebrate	1.800000e-10	1.200e+03	2.000000
63	63	<i>Gastropus hyptopus</i>	invertebrate	1.350000e-10	9.000e+02	2.000000
64	64	<i>Gastropus stylifer</i>	invertebrate	1.264744e-10	2.340e+04	2.000000

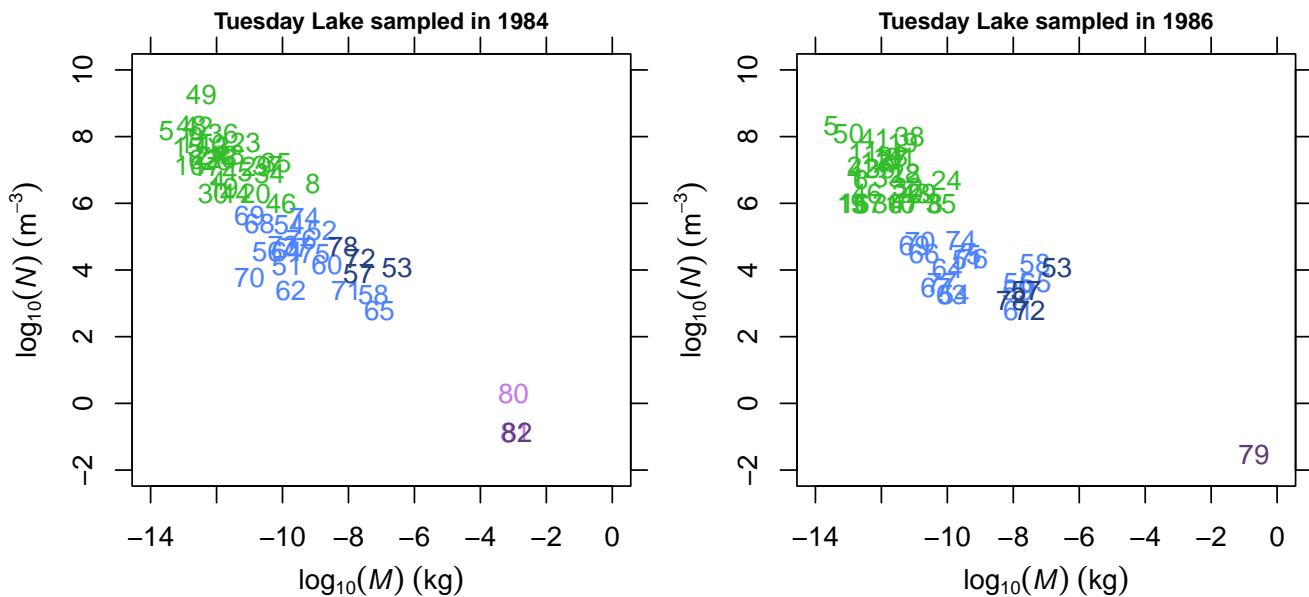
65	65	Holopedium gibberum	invertebrate	5.372500e-08	2.400e+03	2.000000
66	66	Kellicottia bostoniensis	invertebrate	2.000000e-11	1.590e+04	2.000000
67	67	Kellicottia longispina	invertebrate	4.500000e-11	1.500e+03	2.000000
68	68	Kellicottia sp.	invertebrate	2.000000e-11	1.280e+05	2.000000
69	69	Keratella cochlearis	invertebrate	1.000000e-11	2.399e+05	2.000000
70	70	Keratella testudo	invertebrate	1.460317e-11	3.780e+04	2.000000
71	71	Leptodiaptomus siciloides	invertebrate	8.800000e-09	1.200e+03	2.000000
72	72	Orthocyclops modestus	invertebrate	2.313846e-08	1.170e+04	3.166667
73	73	Ploesoma sp.	invertebrate	1.050000e-10	2.790e+04	2.000000
74	74	Polyarthra vulgaris	invertebrate	4.306820e-10	2.258e+05	2.000000
75	75	Synchaeta sp.	invertebrate	6.629293e-10	2.970e+04	2.000000
76	76	Trichocerca cylindrica	invertebrate	4.249133e-10	5.190e+04	2.000000
77	77	Trichocerca multicrinis	invertebrate	2.351765e-10	2.550e+04	2.000000
78	78	Tropocyclops prasinus	invertebrate	6.900000e-09	2.520e+04	3.166667
79	79	Micropterus salmoides	vert.ecto	2.000000e-01	1.500e-02	3.737936
80	80	Phoxinus eos	vert.ecto	1.010000e-03	9.850e-01	3.523756
81	81	Phoxinus neogaeus	vert.ecto	1.170000e-03	6.650e-02	3.523756
82	82	Umbra limi	vert.ecto	1.290000e-03	6.600e-02	3.796484

This code fragment could be easily extended to include any node property that NPS can compute.

### 8.3 Plot IDs

The following code fragment show how to produce a plot of the two communities side by side, showing global IDs.

```
> par(mfrow=c(1,2))
> for(community in TL)
{
  PlotNvM(community, show.nodes.as='labels', show.web=FALSE,
          node.labels=map[NP(community, 'node')], xlim=c(-14, 0),
          ylim=c(-2, 10))
}
```



By default `PlotNvM` highlights species that are cannibals, which are shown in a darker colour. See help for the `PlotNPS` function and the ‘PlotsAndStats’ vignette for more information.

### References

- L.E. Brown, F.K. Edwards, A.M. Milner, G. Woodward, and M.E. Ledger. Food web complexity and allometric scaling relationships in stream mesocosms: implications for experimentation. *Journal of Animal Ecology*, 80(4):884–895, 2011. doi: 10.1111/j.1365-2656.2011.01814.x.
- K. Layer, J.O. Riede, A.G. Hildrew, and G. Woodward. Food web structure and stability in 20 streams across a wide ph gradient. *Advances In Ecological Research*, 42:265–299, 2010. doi: 10.1016/S0065-2504(10)42005-X.
- M.E. Ledger, F.K. Edwards, L.E. Brown, A.M. Milner, and G. Woodward. Impact of simulated drought on ecosystem biomass production: an experimental test in stream mesocosms. *Global Change Biology*, 17(7):2288–2297, 2011. doi: 10.1111/j.1365-2486.2011.02420.x.

- M.E. Ledger, L.E. Brown, F.K. Edwards, A.M. Milner, and G. Woodward. Drought alters the structure and functioning of complex food webs. *Nature Climate Change*, 2(9):1–5, 2012. doi: 10.1038/nclimate1684.
- J.O. Riede, B.C. Rall, C. Banasek-Richter, S.A. Navarrete, E.A. Wieters, M.C. Emmerson, U. Jacob, and U. Brose. Scaling of food-web properties with diversity and complexity across ecosystems. *Advances In Ecological Research*, 42:139–170, 2010. doi: 10.1016/S0065-2504(10)42003-6.
- G. Woodward, L.E. Brown, F. Edwards, L.N. Hudson, A.M. Milner, D.C. Reuman, and M.E. Ledger. Climate change impacts in multispecies systems: drought alters food web size-structure in a field experiment. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1605):2903–2912, 2012. doi: 10.1098/rstb.2012.0232.