

# Package ‘Nonpareil’

January 20, 2025

**Title** Metagenome Coverage Estimation and Projections for 'Nonpareil'

**Type** Package

**Version** 3.5.3

**Author** Luis M. Rodriguez-R [aut, cre]

**Maintainer** Luis M. Rodriguez-R <lmrodriguezr@gmail.com>

**Description** Plot, process, and analyze NPO files produced by  
'Nonpareil' <<http://enve-omics.ce.gatech.edu/nonpareil/>>.

**URL** <http://enve-omics.ce.gatech.edu/nonpareil/>

**Depends** R (>= 2.9)

**Imports** methods

**License** Artistic-2.0

**LazyLoad** yes

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-06-28 15:20:02 UTC

## Contents

+,Nonpareil.Set,ANY-method . . . . .	2
Nonpareil . . . . .	3
Nonpareil.add.curve . . . . .	3
Nonpareil.antif . . . . .	4
Nonpareil.col . . . . .	4
Nonpareil.coverage_factor . . . . .	5
Nonpareil.curve . . . . .	5
Nonpareil.Curve-class . . . . .	7
Nonpareil.curve.batch . . . . .	8
Nonpareil.f . . . . .	9

Nonpareil.fit_model . . . . .	9
Nonpareil.legend . . . . .	10
Nonpareil.read_data . . . . .	10
Nonpareil.read_metadata . . . . .	11
Nonpareil.set . . . . .	11
Nonpareil.Set-class . . . . .	12
plot.Nonpareil.Curve . . . . .	13
plot.Nonpareil.Set . . . . .	14
predict.Nonpareil.Curve . . . . .	15
print.Nonpareil.Curve . . . . .	16
print.Nonpareil.Set . . . . .	16
summary.Nonpareil.Curve . . . . .	17
summary.Nonpareil.Set . . . . .	17
\$.Nonpareil.Curve-method . . . . .	18
\$.Nonpareil.Set-method . . . . .	18
\$<-,Nonpareil.Curve-method . . . . .	19
\$<-,Nonpareil.Set-method . . . . .	19

## Index 20

---

+,Nonpareil.Set,ANY-method  
*Alias of Nonpareil.add.curve.*

---

### Description

Alias of Nonpareil.add.curve.

### Usage

```
## S4 method for signature 'Nonpareil.Set,ANY'
e1 + e2
```

### Arguments

e1            Nonpareil.Set object (nps).  
e2            Nonpareil.Curve object (np).

---

Nonpareil	<i>Nonpareil: Metagenome Coverage Estimation and Projections for 'Nonpareil'.</i>
-----------	---

---

**Description**

Plot, process, and analyze NPO files produced by 'Nonpareil' <http://enve-omics.ce.gatech.edu/nonpareil/>.

**Citation**

If you use Nonpareil, please cite: Rodriguez-R et al. 2018. Nonpareil 3: Fast estimation of metagenomic coverage and sequence diversity. *mSystems* 3(3): e00039-18. DOI: 10.1128/mSystems.00039-18.

Rodriguez-R & Konstantinidis. 2014. Nonpareil: a redundancy-based approach to assess the level of coverage in metagenomic datasets. *Bioinformatics* 30 (5): 629-635. DOI: 10.1093/bioinformatics/btt584.

For an extended discussion on coverage in metagenomic data, see also:

Rodriguez-R & Konstantinidis. 2014. Estimating coverage in metagenomic data sets and why it matters. *The ISME Journal* 8: 2349–2351. DOI: 10.1038/ismej.2014.76.

**Author(s)**

**Maintainer:** Luis M. Rodriguez-R <lmrodriguezr@gmail.com>

**See Also**

Useful links:

- <http://enve-omics.ce.gatech.edu/nonpareil/>

---

Nonpareil.add.curve	<i>Adds a Nonpareil.Curve to a Nonpareil.Set.</i>
---------------------	---

---

**Description**

Adds a Nonpareil.Curve to a Nonpareil.Set.

**Usage**

```
Nonpareil.add.curve(nps, np)
```

**Arguments**

nps	Nonpareil.Set object.
np	Nonpareil.Curve object.

**Value**

Returns the Nonpareil.Set including the newly added Nonpareil.Curve.

---

Nonpareil.antif	<i>Complement function of Nonpareil.f.</i>
-----------------	--

---

**Description**

Complement function of Nonpareil.f.

**Usage**

Nonpareil.antif(y, a, b)

**Arguments**

y	Values of abundance-weighted average coverage.
a	Parameter alpha of the gamma CDF.
b	Parameter beta of the gamma CDF.

**Value**

Estimated sequencing effort.

---

Nonpareil.col	<i>Returns the color of the curve.</i>
---------------	--

---

**Description**

Returns the color of the curve.

**Usage**

Nonpareil.col(x, alpha = 1)

**Arguments**

x	Nonpareil.Curve or Nonpareil.Set object.
alpha	Alpha level of the color from 0 to 1.

---

`Nonpareil.coverage_factor`*Factor to transform redundancy into coverage (internal function).*

---

**Description**

Factor to transform redundancy into coverage (internal function).

**Usage**

```
Nonpareil.coverage_factor(x)
```

**Arguments**

`x` Nonpareil.Curve object.

**Value**

A numeric scalar.

---

`Nonpareil.curve`*Generates a Nonpareil curve from an .npo file*

---

**Description**

Generates a Nonpareil curve from an .npo file

**Usage**

```
Nonpareil.curve(  
  file,  
  plot = TRUE,  
  label = NA,  
  col = NA,  
  enforce.consistency = TRUE,  
  star = 95,  
  correction.factor = TRUE,  
  weights.exp = NA,  
  skip.model = FALSE,  
  ...  
)
```

**Arguments**

<code>file</code>	Path to the .npo file, containing the read redundancy.
<code>plot</code>	Determines if the plot should be produced. If FALSE, it computes the coverage and the model without plotting.
<code>label</code>	Name of the dataset. If NA, it is determined by the file name.
<code>col</code>	Color of the curve. If NA, a random color is assigned (even if <code>plot = FALSE</code> ).
<code>enforce.consistency</code>	If TRUE, it fails verbosely on insufficient data, otherwise it warns about the inconsistencies and attempts the estimations.
<code>star</code>	Objective coverage in percentage; i.e., coverage value considered near-complete.
<code>correction.factor</code>	Should the overlap-dependent (or kmer-length-dependent) correction factor be applied? If FALSE, redundancy is assumed to equal coverage.
<code>weights.exp</code>	Vector of values to be tested (in order) as exponent of the weights distribution. If the model fails to converge, sometimes manual modifications in this parameter may help. By default (NA), five different values are tested in the following order: For linear sampling, -1.1, -1.2, -0.9, -1.3, -1. For logarithmic sampling (-d option in Nonpareil), 0, 1, -1, 1.3, -1.1, 1.5, -1.5.
<code>skip.model</code>	If set, skips the model estimation altogether.
<code>...</code>	Any additional parameters passed to <code>plot.Nonpareil.Curve</code> .

**Value**

Returns invisibly a `Nonpareil.Curve` object

**Examples**

```
# Generate a Nonpareil plot
file <- system.file("extdata", "LakeLanier.npo", package = "Nonpareil")
np <- Nonpareil.curve(file)

# Produce the same plot but using powers of 1,000bp as X axis labels
Nonpareil.curve(file, xaxt = "n", xlab = "Sequencing Effort")
axis(
  1L, at = 10L^seq(3L, 12L, by = 3L),
  labels = paste(1L, c("Kbp", "Mbp", "Gbp", "Tbp"))
)

# Show the estimated values
print(np)

# Predict coverage for 20Gbp
predict(np, 20e9)

# Obtain the Nd diversity index
np$diversity
```

---

Nonpareil.Curve-class *A single Nonpareil curve. This object can be produced by Nonpareil.curve and supports S4 methods plot, summary, print, and predict. For additional details, see help for summary.Nonpareil.Curve.*

---

### Description

A single Nonpareil curve. This object can be produced by `Nonpareil.curve` and supports S4 methods `plot`, `summary`, `print`, and `predict`. For additional details, see `help` for `summary.Nonpareil.Curve`.

### Slots

`file` Input .npo file.  
`label` Name of the dataset.  
`col` Color of the dataset.  
`L` Read length.  
`AL` Adjusted read length (same as `L` for alignment).  
`R` Number of reads.  
`LR` Effective sequencing effort used.  
`overlap` Minimum read overlap.  
`ksize` K-mer size (for kmer kernel only).  
`log.sample` Multiplier of the log-sampling (or zero if linear).  
`kernel` Read-comparison kernel.  
`version` Nonpareil version used.  
`x.obs` Rarefied sequencing effort.  
`x.adj` Adjusted rarefied sequencing effort.  
`y.red` Rarefied redundancy (observed).  
`y.cov` Rarefied coverage (corrected).  
`y.sd` Standard deviation of rarefied coverage.  
`y.p25` Percentile 25 (1st quartile) of rarefied coverage.  
`y.p50` Percentile 50 (median) of rarefied coverage.  
`y.p75` Percentile 75 (3rd quartile) of rarefied coverage.  
`kappa` Dataset redundancy.  
`C` Dataset coverage.  
`consistent` Is the data sufficient for accurate estimation?  
`star` Coverage considered 'nearly complete'.  
`has.model` Was the model successfully estimated?  
`warning` Warnings generated on consistency or model fitting.

LRstar Projected seq. effort for nearly complete coverage.

modelR Pearson's R for the estimated model.

diversity Dataset Nd index of sequence diversity.

model Fitted sigmoidal model.

call Call producing this object.

---

Nonpareil.curve.batch *Alias of* Nonpareil.set.

---

## Description

Alias of Nonpareil.set.

## Usage

```
Nonpareil.curve.batch(
  files,
  col = NA,
  labels = NA,
  plot = TRUE,
  plot.opts = list(),
  ...
)
```

## Arguments

files	Vector with the paths to the .npo files.
col	Color of the curves (vector). If not passed, values are randomly assigned. Values are recycled.
labels	Labels of the curves (vector). If not passed, values are determined by the file-name. Values are recycled.
plot	If TRUE, it generates the Nonpareil curve plots.
plot.opts	Any parameters accepted by plot.Nonpareil. Set as a list.
...	Any additional parameters accepted by Nonpareil.curve.



---

Nonpareil.f	<i>Function of the projected model.</i>
-------------	---

---

**Description**

Function of the projected model.

**Usage**

```
Nonpareil.f(x, a, b)
```

**Arguments**

x	Values of sequencing effort (in bp).
a	Parameter alpha of the Gamma CDF.
b	Parameter beta of the Gamma CDF.

**Value**

Predicted values of abundance-weighted average coverage.

---

Nonpareil.fit_model	<i>Fit the sigmoidal model to the rarefied coverage.</i>
---------------------	--

---

**Description**

Fit the sigmoidal model to the rarefied coverage.

**Usage**

```
Nonpareil.fit_model(np, weights.exp)
```

**Arguments**

np	Nonpareil.Curve object.
weights.exp	Numeric; see Nonpareil.curve for details.

---

Nonpareil.legend      *Generates a legend for Nonpareil plots.*

---

### Description

Generates a legend for Nonpareil plots.

### Usage

```
Nonpareil.legend(np, x, y = 0.3, ...)
```

### Arguments

np	A Nonpareil.Set or Nonpareil.Curve object, or a list of Nonpareil.Curve objects.
x	X coordinate, or any character string accepted by legend (e.g., 'bottomright').
y	Y coordinate.
...	Any other parameters supported by legend().

### Value

Returns invisibly a list, same as legend.

---

Nonpareil.read\_data      *Read the data tables and extract direct estimates.*

---

### Description

Read the data tables and extract direct estimates.

### Usage

```
Nonpareil.read_data(x, correction.factor)
```

### Arguments

x	Nonpareil.Curve object.
correction.factor	Logical; see Nonpareil.curve for details.

---

```
Nonpareil.read_metadata
```

*Read the metadata headers.*

---

**Description**

Read the metadata headers.

**Usage**

```
Nonpareil.read_metadata(x)
```

**Arguments**

x                    Nonpareil.Curve object.

---

```
Nonpareil.set
```

*Generates a collection of Nonpareil curves (a Nonpareil.Set object) and (optionally) plots all of them in a single canvas.*

---

**Description**

Generates a collection of Nonpareil curves (a Nonpareil.Set object) and (optionally) plots all of them in a single canvas.

**Usage**

```
Nonpareil.set(
  files,
  col = NA,
  labels = NA,
  plot = TRUE,
  plot.opts = list(),
  ...
)
```

**Arguments**

files                Vector with the paths to the .npo files.

col                  Color of the curves (vector). If not passed, values are randomly assigned. Values are recycled.

labels               Labels of the curves (vector). If not passed, values are determined by the file-name. Values are recycled.

plot                 If TRUE, it generates the Nonpareil curve plots.

plot.opts            Any parameters accepted by plot.Nonpareil.Set as a list.

...                  Any additional parameters accepted by Nonpareil.curve.

**Value**

Returns invisibly a `Nonpareil.Set` object.

**Examples**

```
# Generate a Nonpareil plot with multiple curves
files <- system.file(
  "extdata",
  c("HumanGut.npo", "LakeLanier.npo", "IowaSoil.npo"),
  package = "Nonpareil"
)
col <- c("orange", "darkcyan", "firebrick4")
nps <- Nonpareil.set(
  files, col = col,
  plot.opts = list(plot.observed = FALSE, model.lwd = 2)
)

# Show the estimated values
print(nps)

# Show current coverage (as %)
summary(nps)[, "C"] * 100

# Extract Nd diversity index
summary(nps)[, "diversity"]

# Extract sequencing effort for nearly complete coverage (in Gbp)
summary(nps)[, "LRstar"] / 1e9

# Predict coverage for a sequencing effort of 10Gbp
sapply(nps$np.curves, predict, 10e9)
```

---

<code>Nonpareil.Set-class</code>	<i>Collection of Nonpareil.Curve objects. This object can be produced by <code>Nonpareil.curve.batch</code> and supports S4 methods <code>plot</code>, <code>summary</code>, and <code>print</code>.</i>
----------------------------------	--

---

**Description**

Collection of `Nonpareil.Curve` objects. This object can be produced by `Nonpareil.curve.batch` and supports S4 methods `plot`, `summary`, and `print`.

**Slots**

`np.curves` List of `Nonpareil.Curve` objects.  
`call` Call producing this object.

---

plot.Nonpareil.Curve *Plot a Nonpareil.Curve object.*

---

### Description

Plot a Nonpareil.Curve object.

### Usage

```
## S3 method for class 'Nonpareil.Curve'
plot(
  x,
  col = NA,
  add = FALSE,
  new = !add,
  plot.observed = TRUE,
  plot.model = TRUE,
  plot.dispersion = FALSE,
  plot.diversity = TRUE,
  xlim = c(1000, 1e+13),
  ylim = c(1e-06, 1),
  main = paste("Nonpareil Curve for", x$label),
  xlab = "Sequencing effort (bp)",
  ylab = "Estimated Average Coverage",
  curve.lwd = 2,
  curve.alpha = 0.4,
  model.lwd = 1,
  model.alpha = 1,
  log = "x",
  arrow.length = 0.05,
  arrow.head = arrow.length,
  ...
)
```

### Arguments

x	Nonpareil.Curve object to plot.
col	Color of the curve. If passed, it overrides the colors set in the Nonpareil.Curve object.
add	If TRUE, it attempts to use a previous (active) canvas to plot the curve.
new	Inverse of 'add'.
plot.observed	Indicates if the observed (rarefied) coverage is to be plotted.
plot.model	Indicates if the fitted model is to be plotted.
plot.dispersion	Indicates if (and how) dispersion of the replicates should be plotted. Supported values are:

	<ul style="list-style-type: none"> <li>• FALSE: no dispersion is plotted (default),</li> <li>• 'sd': one standard deviation around the mean,</li> <li>• 'ci95': 95</li> <li>• 'ci90': 90</li> <li>• 'ci50': 50</li> <li>• 'iq': Inter-quartile range.</li> </ul>
plot.diversity	If TRUE, the diversity estimate is plotted as a small arrow below the Nonpareil curve.
xlim	Limits of the sequencing effort (X-axis).
ylim	Limits of the coverage (Y-axis).
main	Title of the plot.
xlab	Label of the X-axis.
ylab	Label of the Y-axis.
curve.lwd	Line width of the rarefied coverage.
curve.alpha	Alpha value (from 0 to 1) of the rarefied coverage.
model.lwd	Line width of the model.
model.alpha	Alpha value (from 0 to 1) of the model.
log	Axis to plot in logarithmic scale. Supported values are: <ul style="list-style-type: none"> <li>• 'x': sequencing effort (default),</li> <li>• 'y': coverage,</li> <li>• 'xy': both logarithmic, or</li> <li>• '': both linear.</li> </ul>
arrow.length	If plot.diversity = TRUE, it determines the length of the arrow to display the diversity (as a fraction of the ylim range).
arrow.head	If plot.diversity = TRUE, it determines the length of the arrow head to display the diversity index (in inches).
...	Additional graphical parameters.

**Value**

Returns invisibly a Nonpareil.Curve object (same as x input). For additional details see help for summary.Nonpareil.Curve.

---

plot.Nonpareil.Set      *Plot a Nonpareil.Set object.*

---

**Description**

Plot a Nonpareil.Set object.

**Usage**

```
## S3 method for class 'Nonpareil.Set'
plot(
  x,
  col = NA,
  labels = NA,
  main = "Nonpareil Curves",
  legend.opts = list(),
  ...
)
```

**Arguments**

x	Nonpareil.Set object to plot.
col	Color of the curves (vector). If passed, it overrides the colors set in the Nonpareil.Curve objects. Values are recycled.
labels	Labels of the curves (vector). If passed, it overrides the labels set in the Nonpareil.Curve objects. Values are recycled.
main	Title of the plot.
legend.opts	Any additional parameters passed to Nonpareil.legend. If FALSE, the legend is not displayed.
...	Any additional parameters passed to plot.Nonpareil.Curve.

**Value**

Returns invisibly a Nonpareil.Set object (same as x input).

---

predict.Nonpareil.Curve

*Predict the coverage for a given sequencing effort.*

---

**Description**

Predict the coverage for a given sequencing effort.

**Usage**

```
## S3 method for class 'Nonpareil.Curve'
predict(object, lr = object$LR, ...)
```

**Arguments**

object	Nonpareil.Curve object.
lr	Sequencing effort for the prediction (in bp).
...	Additional parameters ignored.

**Value**

Returns the expected coverage at the given sequencing effort.

---

`print.Nonpareil.Curve` *Prints and returns invisibly a summary of the Nonpareil.Curve results.*

---

**Description**

Prints and returns invisibly a summary of the `Nonpareil.Curve` results.

**Usage**

```
## S3 method for class 'Nonpareil.Curve'
print(x, ...)
```

**Arguments**

<code>x</code>	<code>Nonpareil.Set</code> object.
<code>...</code>	Additional parameters ignored.

**Value**

Returns the summary invisibly. See help for `summary.Nonpareil.Curve` for additional information.

---

`print.Nonpareil.Set` *Prints and returns invisibly a summary of the Nonpareil.Set results.*

---

**Description**

Prints and returns invisibly a summary of the `Nonpareil.Set` results.

**Usage**

```
## S3 method for class 'Nonpareil.Set'
print(x, ...)
```

**Arguments**

<code>x</code>	<code>Nonpareil.Set</code> object.
<code>...</code>	Additional parameters ignored.

**Value**

Returns the summary invisibly. See help for `summary.Nonpareil.Curve` and `summary.Nonpareil.Set` for additional information.



---

```
summary.Nonpareil.Curve
```

*Returns a summary of the Nonpareil.Curve results.*

---

### Description

Returns a summary of the Nonpareil.Curve results.

### Usage

```
## S3 method for class 'Nonpareil.Curve'  
summary(object, ...)
```

### Arguments

object            Nonpareil.Curve object.  
...                Additional parameters ignored.

### Value

Returns a matrix with the following values for the dataset:

- kappa: "Redundancy" value of the entire dataset.
- C: Average coverage of the entire dataset.
- LRstar: Estimated sequencing effort required to reach the objective average coverage (star, 95
- LR: Actual sequencing effort of the dataset.
- modelR: Pearson's R coefficient between the rarefied data and the projected model.
- diversity: Nonpareil sequence-diversity index (Nd). This value's units are the natural logarithm of the units of sequencing effort (log-bp), and indicates the inflection point of the fitted model for the Nonpareil curve. If the fit doesn't converge, or the model is not estimated, the value is zero (0).

---

```
summary.Nonpareil.Set
```

*Returns a summary of the Nonpareil.Set results.*

---

### Description

Returns a summary of the Nonpareil.Set results.

### Usage

```
## S3 method for class 'Nonpareil.Set'  
summary(object, ...)
```

**Arguments**

object	Nonpareil.Set object.
...	Additional parameters ignored.

**Value**

Returns a matrix with different values for each dataset. For additional details on the values returned, see help for `summary.Nonpareil.Curve`.

---

*\$.Nonpareil.Curve-method*

*Get attribute.*

---

**Description**

Get attribute.

**Usage**

```
## S4 method for signature 'Nonpareil.Curve'
x$name
```

**Arguments**

x	Nonpareil.Curve object.
name	Attribute.

---

*\$.Nonpareil.Set-method*

*Get attribute.*

---

**Description**

Get attribute.

**Usage**

```
## S4 method for signature 'Nonpareil.Set'
x$name
```

**Arguments**

x	Nonpareil.Set object.
name	Attribute.

---

*\$<- ,Nonpareil.Curve-method*  
*Set attribute.*

---

**Description**

Set attribute.

**Usage**

```
## S4 replacement method for signature 'Nonpareil.Curve'  
x$name <- value
```

**Arguments**

x	Nonpareil.Curve object.
name	Attribute.
value	New value.

---

*\$<- ,Nonpareil.Set-method*  
*Set attribute.*

---

**Description**

Set attribute.

**Usage**

```
## S4 replacement method for signature 'Nonpareil.Set'  
x$name <- value
```

**Arguments**

x	Nonpareil.Set object.
name	Attribute.
value	New value.

# Index

`+`, `Nonpareil.Set`, ANY-method, 2  
`$`, `Nonpareil.Curve`-method, 18  
`$`, `Nonpareil.Set`-method, 18  
`$<-`, `Nonpareil.Curve`-method, 19  
`$<-`, `Nonpareil.Set`-method, 19

`Nonpareil`, 3  
`Nonpareil-package` (`Nonpareil`), 3  
`Nonpareil.add.curve`, 3  
`Nonpareil.antif`, 4  
`Nonpareil.col`, 4  
`Nonpareil.coverage_factor`, 5  
`Nonpareil.curve`, 5  
`Nonpareil.Curve`-class, 7  
`Nonpareil.curve.batch`, 8  
`Nonpareil.f`, 9  
`Nonpareil.fit_model`, 9  
`Nonpareil.legend`, 10  
`Nonpareil.read_data`, 10  
`Nonpareil.read_metadata`, 11  
`Nonpareil.set`, 11  
`Nonpareil.Set`-class, 12

`plot.Nonpareil.Curve`, 13  
`plot.Nonpareil.Set`, 14  
`predict.Nonpareil.Curve`, 15  
`print.Nonpareil.Curve`, 16  
`print.Nonpareil.Set`, 16

`summary.Nonpareil.Curve`, 17  
`summary.Nonpareil.Set`, 17