

# Package: deeppicker (via r-universe)

May 13, 2026

**Type** Package

**Title** R Interface to DEEP Picker for One- and Two-Dimensional Peak Picking

**Version** 0.5

**Description** Provides an R interface to the DEEP Picker neural-network peak-picking tools for one- and two-dimensional NMR spectra. The package wraps the peak-picking engines, returning peak positions, intensities, line-shape parameters, and confidence scores. It also includes readers for supported spectrum formats, including NMRPipe, Sparky, TopSpin ASCII, JSON, CSV, and ldw files, so spectra can be imported and processed directly.

**License** GPL-3

**URL** <https://github.com/smith-group/deeppicker>,  
<https://smith-group.github.io/deeppicker/>

**Encoding** UTF-8

**Depends** R (>= 3.6.0)

**Imports** Rcpp

**LinkingTo** Rcpp, RcppEigen

**Suggests** testthat (>= 3.0.0)

**SystemRequirements** C++17

**Config/testthat/edition** 3

**RoxygenNote** 7.3.3

**Repository** <https://smith-group.r-universe.dev>

**Date/Publication** 2026-03-13 18:50:54 UTC

**RemoteUrl** <https://github.com/smith-group/deeppicker>

**RemoteRef** HEAD

**RemoteSha** 4ca92958f3170b57e83b902fffa99bcd0f2d84fc

## Contents

deep_picker . . . . .	2
deep_picker_1d . . . . .	4
read_spectrum_1d . . . . .	7
read_spectrum_2d . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

deep_picker	<i>Run DEEP Picker on a 2D NMR Spectrum</i>
-------------	---

---

## Description

DEEP Picker is an artificial neural network based 2D NMR spectral peak picking and deconvolution tool. It predicts every 2D cross-peak locally without taking into account the behavior of spectral data points that are further away. In practice, it provides an excellent starting point for downstream quantitative fitting workflows.

## Usage

```

deep_picker(
  spectrum,
  ppm = NULL,
  noise = NULL,
  scale = 5.5,
  scale2 = 3,
  scale_negative = scale,
  scale2_negative = scale2,
  model = 1L,
  auto_ppp = TRUE,
  t1_noise = FALSE,
  negative = FALSE,
  debug_flag = 0L,
  verbose = TRUE,
  as_data_frame = TRUE
)

```

## Arguments

spectrum	Numeric matrix with the indirect dimension in rows and the direct dimension in columns.
ppm	Optional numeric vector 'c(begin1, step1, begin2, step2)' giving the ppm origin and increment for the direct and indirect dimensions. If 'NULL', 'deep_picker()' derives these values from numeric column names (direct dimension) and row names (indirect dimension).

noise	Spectrum noise level. If 'NULL', 'deep_picker()' estimates the noise internally using the same default variance-based procedure as the DEEP Picker command-line program.
scale	Minimal peak amplitude cutoff, expressed as a multiple of the noise level. The default is '5.5'.
scale2	Noise-floor cutoff, expressed as a multiple of the noise level. Spectral points below this threshold are set to zero before peak picking. The default is '3.0'.
scale_negative	Minimal negative-peak amplitude cutoff, expressed as a multiple of the noise level. The default is 'scale'.
scale2_negative	Negative-peak noise-floor cutoff, expressed as a multiple of the noise level. The default is 'scale2'.
model	ANN model selection. '1L' corresponds to the broader PPP range typical for protein spectra; '2L' corresponds to the narrower PPP range typical for metabolomics spectra.
auto_ppp	Logical; whether to adjust PPP automatically using cubic spline interpolation.
t1_noise	Logical; whether to remove possible 't1' noise peaks.
negative	Logical; whether to pick negative peaks in addition to positive peaks.
debug_flag	Integer debug flag forwarded to the DEEP Picker core.
verbose	Logical; whether to print DEEP Picker progress messages.
as_data_frame	Logical; if 'TRUE', return a data frame of peaks. If 'FALSE', return a list with the peak table, estimated median widths, and noise level.

## Details

This wrapper exposes the DEEP Picker 2D peak-picking engine for spectra that are already available in R memory. The 'model' argument follows the DEEP Picker guidance: use '1L' for spectra with about 6 to 20 points per peak (typical for protein spectra) and '2L' for spectra with about 4 to 12 points per peak (typical for metabolomics spectra). When 'auto\_ppp = TRUE', DEEP Picker adjusts PPP automatically using cubic spline interpolation; the DEEP Picker documentation recommends leaving this enabled unless there is a specific reason not to.

## Value

If 'as\_data\_frame = TRUE', a data frame with one row per picked peak and the following columns:

'x', 'y' Peak coordinates in DEEP Picker point units for the direct ('x') and indirect ('y') dimensions.

'ppm\_x', 'ppm\_y' Peak positions converted to ppm for the direct and indirect dimensions.

'intensity' Estimated peak intensity.

'sigmax', 'sigmay' Estimated Gaussian width components in the direct and indirect dimensions.

'gammax', 'gammay' Estimated Lorentzian width components in the direct and indirect dimensions.

'confidence' Peak confidence score on a 0 to 1 scale. For 2D peaks, this wrapper reports the smaller of the two axis-specific confidence values returned by DEEP Picker.

If `'as_data_frame = FALSE'`, a list with components:

- 'peaks'** The peak data frame described above.
- 'median\_width'** Length-2 numeric vector of median peak widths in DEEP Picker point units, ordered as direct dimension then indirect dimension.
- 'noise\_level'** Noise level used during picking, either supplied by the user or estimated internally.

## References

Li, D.-W., Hansen, A. L., Yuan, C., Bruschweiler-Li, L., and Bruschweiler, R. (2021). DEEP Picker is a Deep Neural Network for Accurate Deconvolution of Complex Two-Dimensional NMR Spectra. *Nature Communications*, 12, 5229. DOI: 10.1038/s41467-021-25496-5.

## Examples

```
path <- system.file("extdata", "mfap.ft2", package = "deeppicker")
spectrum <- read_spectrum_2d(path)
peaks <- deep_picker(spectrum)

ppm_x <- as.numeric(colnames(spectrum))
ppm_y <- as.numeric(rownames(spectrum))
ix <- order(ppm_x)
iy <- order(ppm_y)
x_inc <- ppm_x[ix]
y_inc <- ppm_y[iy]
z <- t(spectrum[iy, ix, drop = FALSE])

positive <- spectrum[spectrum > 0]
base_level <- 6 * stats::median(abs(positive))
max_level <- max(positive)
levels <- exp(seq(log(base_level), log(max_level), length.out = 8))

contour(x_inc, y_inc, z, levels = levels, drawlabels = FALSE, lwd = 0.5,
        xlim = rev(range(x_inc)), ylim = rev(range(y_inc)),
        xlab = "1H (ppm)", ylab = "15N (ppm)",
        main = "DEEP Picker Example: mfap.ft2")

cols <- grDevices::hcl.colors(100, "YlOrRd", rev = TRUE)
conf_idx <- pmax(1L, ceiling(peaks$confidence * 99))
points(peaks$ppm_x, peaks$ppm_y, pch = 16, cex = 0.35, col = cols[conf_idx])
legend("topright",
       legend = c("Lower Confidence", "Higher Confidence"),
       pch = 16, pt.cex = 0.35, col = cols[c(25, 95)], bty = "n")
```

## Description

DEEP Picker uses artificial neural networks for NMR peak picking and deconvolution. This wrapper exposes the 1D DEEP Picker engine for spectra that are already available in R memory.

## Usage

```
deep_picker_1d(  
  spectrum,  
  ppm = NULL,  
  noise = NULL,  
  scale = 5.5,  
  scale2 = 3,  
  model = 2L,  
  auto_ppp = TRUE,  
  interp_step = 1,  
  negative = FALSE,  
  verbose = TRUE  
)
```

## Arguments

spectrum	Numeric vector containing the 1D spectrum intensities.
ppm	Optional ppm information. Supply either a numeric vector of length 3, 'c(begin, step, stop)', or a numeric vector the same length as 'spectrum'. If 'NULL', 'deep_picker_1d()' derives ppm values from 'names(spectrum)'.
noise	Spectrum noise level. If 'NULL', 'deep_picker_1d()' estimates noise internally using the same default variance-based procedure as the command-line program.
scale	Minimal peak amplitude cutoff, expressed as a multiple of the noise level. The default is '5.5'.
scale2	Signal-region cutoff, expressed as a multiple of the noise level. The default is '3.0'.
model	ANN model selection. '1L' corresponds to the broader PPP range; '2L' corresponds to the narrower PPP range used by the 1D command-line tool by default.
auto_ppp	Logical; whether to adjust PPP automatically using cubic spline interpolation.
interp_step	Numeric interpolation step. '1' means no interpolation. Values other than '1' suppress automatic PPP adjustment.
negative	Logical; whether to pick negative peaks in addition to positive peaks.
verbose	Logical; whether to print DEEP Picker progress messages.

## Details

The 'model' argument follows the DEEP Picker guidance: use '1L' for broader peaks and '2L' for narrower peaks. When 'auto\_ppp = TRUE', DEEP Picker adjusts PPP automatically using cubic spline interpolation. As in the command-line program, automatic PPP adjustment is disabled when 'interp\_step' is not '1'.

**Value**

A data frame with one row per picked peak and the following columns:

- ‘**x**‘ Peak coordinate in DEEP Picker point units.
- ‘**ppm**‘ Peak position in ppm.
- ‘**height**‘ Estimated peak height.
- ‘**sigmax**‘ Estimated Gaussian width component.
- ‘**gammax**‘ Estimated Lorentzian width component.
- ‘**volume**‘ Estimated integrated peak volume.
- ‘**confidence**‘ Peak confidence score on a 0 to 1 scale.

**References**

Li, D.-W., Hansen, A. L., Yuan, C., Bruschweiler-Li, L., and Bruschweiler, R. (2021). DEEP Picker is a Deep Neural Network for Accurate Deconvolution of Complex Two-Dimensional NMR Spectra. *Nature Communications*, 12, 5229. DOI: 10.1038/s41467-021-25496-5.

**Examples**

```
path <- system.file("extdata", "tyrosine.ft1", package = "deeppicker")
spectrum <- read_spectrum_1d(path)
peaks <- deep_picker_1d(spectrum, scale = 100)

ppm <- as.numeric(names(spectrum))
plot(ppm, spectrum, type = "l", xlim = c(7.5, -0.5),
      xlab = "1H (ppm)", ylab = "Intensity",
      main = "DEEP Picker 1D Example: tyrosine.ft1")

cols <- grDevices::hcl.colors(100, "YlOrRd", rev = TRUE)
conf_idx <- pmax(1L, ceiling(peaks$confidence * 99))
points(peaks$ppm, peaks$height, pch = 16, cex = 0.5, col = cols[conf_idx])
legend("topright",
       legend = c("Lower Confidence", "Higher Confidence"),
       pch = 16, pt.cex = 0.5, col = cols[c(25, 95)], bty = "n")

centers <- c(6.73, 4.78, 3.36, 2.72, 1.84, -0.05)
half_width <- 0.25
old_par <- par(mfrow = c(3, 2))
for (center in centers) {
  xlim <- c(center + half_width, center - half_width)
  in_window <- ppm >= min(xlim) & ppm <= max(xlim)
  ylim <- range(spectrum[in_window], finite = TRUE)
  plot(ppm, spectrum, type = "l", xlim = xlim, ylim = ylim,
       xlab = "1H (ppm)", ylab = "Intensity",
       main = sprintf("Window at %.2f ppm", center))
  points(peaks$ppm, peaks$height, pch = 16, cex = 0.5, col = cols[conf_idx])
}
par(old_par)
```

---

read_spectrum_1d	<i>Read a 1D Spectrum with DEEP Picker Parsers</i>
------------------	--

---

### Description

Reads a 1D spectrum file using the file parsers bundled with DEEP Picker and returns the real spectrum as a named numeric vector. The names are the ppm values parsed from the file.

### Usage

```
read_spectrum_1d(path, verbose = TRUE)
```

### Arguments

path	Path to a 1D spectrum file.
verbose	Logical; whether to print parser progress messages.

### Details

Supported formats are chosen by file extension and include NMRPipe (‘.ft1’), text (‘.txt’), CSV (‘.csv’), JSON (‘.json’), Sparky (‘.ucsf’), and the DEEP Picker ‘.ldw’ format.

### Value

A named numeric vector containing the real spectrum. The names are numeric ppm values stored as character strings.

### Examples

```
path <- system.file("extdata", "tyrosine.ft1", package = "deeppicker")
x <- read_spectrum_1d(path)
str(x)
```

---

read_spectrum_2d	<i>Read a 2D Spectrum with DEEP Picker Parsers</i>
------------------	--

---

### Description

Reads a 2D spectrum file using the file parsers bundled with DEEP Picker and returns the real spectrum as a numeric matrix. Row names and column names are the ppm values parsed from the file for the indirect and direct dimensions.

### Usage

```
read_spectrum_2d(path, verbose = TRUE)
```

**Arguments**

<code>path</code>	Path to a 2D spectrum file.
<code>verbose</code>	Logical; whether to print parser progress messages.

**Details**

Supported formats are chosen by file extension and include NMRPipe (`.ft2`), text (`.txt`), CSV (`.csv`), JSON (`.json`), Sparky (`.ucsf`), and the DEEP Picker `.ldw` format.

**Value**

A numeric matrix with row names and column names set to ppm values.

**Examples**

```
path <- system.file("extdata", "mfap.ft2", package = "deeppicker")
x <- read_spectrum_2d(path)
str(x)
```

# Index

`deep_picker`, [2](#)

`deep_picker_1d`, [4](#)

`read_spectrum_1d`, [7](#)

`read_spectrum_2d`, [7](#)