

# Introduction to the atomicEmission Package

2018-09-01

This package includes functions to simulate the visible atomic emission spectra for 46 elements, between 390 nm and 700 nm, and to display the result of the simulation as an emission spectrum (a plot of absolute or relative intensity as a function of wavelength), or as an emission line spectrum or absorbance line spectrum (showing the wavelengths where light is emitted or absorbed).

## What Elements Are Available?

Data is included for 46 elements with atomic numbers 1–32, 36–38, 47–50, 54,–56, 74, 79, 80, and 82, each as a neutral gas phase atom. This list includes all elements in the first four rows of the periodic table—with the exceptions of germanium, arsenic, selenium, and bromine, which either do not have emission lines between 390 nm and 700 nm, or do not have sufficient information to determine emission intensities—as well as the alkali metals, alkaline earths, and inert gases in rows five and six of the periodic table, and a few additional elements of interest (Ag, Cd, In, Sn, W, Au, Hg, and Pb).

## How Are Functions Named?

The functions in this package take the general form **actionOutput**, where **action** explains what the function does and **Output** indicates the result that is returned. The available actions are **simulate**, **plot**, **identify**, and **create**, and the available results are **AES** for atomic emission spectrum, **Peaks** for the peaks in an emission spectrum, **Lines** for the lines in an emission or absorbance spectrum, and **Vis** for a visible spectrum. Table 1 provides a summary of the available functions and examples of their use are shown in the sections that follow.

Table 1: Functions in the **atomicEmission** package

Function	Purpose
<b>simulateAES</b>	simulates an element's atomic emission spectrum
<b>plotAES</b>	plots emission intensity as function of wavelength
<b>plotLines</b>	displays the emission or the absorbance line spectrum
<b>identifyPeaks</b>	creates a table of the most intense emission peaks
<b>createVis</b>	converts visible wavelengths to their corresponding colors; this is a helper function called by <b>plotLines</b> and is not discussed further in this document (see the function's help file for additional information)

## Simulating an Atomic Emission Spectrum

The format for the function **simulateAES** is

```
simulateAES(atom = "H", temperature = 7000, width = 0.3)
```

with its arguments and their default values defined below:

- **atom**: The element of interest, which is entered in one of three ways: using the element's atomic symbol, **atom** = "H"; using the element's name, **atom** = "hydrogen"; or using the element's atomic number, **atom** = 1. Defaults to hydrogen.
- **temperature**: The temperature, in K, with a lower limit of 1000 K and an upper limit of 12000 K. Defaults to 7000 K. The choice of temperature determines the model used to simulation the spectrum; see the section on Computational Details for more information).

- **width**: The total width of an emission peak, in nm. Defaults to 0.3 nm.

To use the `simulateAES` function, we assign it to an object and pass along values for the function's arguments. If we wish to accept the function's default values, then we enter the following line of code, where `hydrogen` is the object that will hold the simulation's results.

```
hydrogen = simulateAES()
```

If we wish to simulate the atomic emission spectrum for chromium at a temperature of 5000 K and with a total peak width of 0.1 nm, then we enter the following line of code

```
chromium = simulateAES(atom = "Cr", temperature = 5000, width = 0.1)
```

Once created, we can use R's structure command, `str()`, to examine the information that is stored within the object as a list:

```
str(chromium)
```

```
## List of 11
## $ element      : chr "chromium"
## $ symbol       : chr "Cr"
## $ temperature  : num 5000
## $ wavelength   : num [1:31001] 390 390 390 390 390 ...
## $ absolute_intensity: num [1:31001] 0.0897 0.0946 0.0999 0.1058 0.1124 ...
## $ relative_intensity: num [1:31001] 0.000191 0.000201 0.000213 0.000225 0.000239 ...
## $ qt          : num 10.4
## $ model        : chr "deGalan"
## $ width        : num 0.1
## $ peaks        : num [1:391] 390 390 391 392 392 ...
## $ peak_max     : num [1:391] 1.77 0.562 4.192 1.031 7.658 ...
```

In addition to returning the function's inputs, the list also includes the wavelengths, the absolute and relative emission intensities at each wavelength, the temperature-dependent partition function and the model used to calculate its value (see the section on computational details for more on this), and the individual emission lines and their emission intensities.

## Plotting an Atomic Emission Spectrum

The `plotAES` function is used to view the atomic emission spectrum; it takes the form

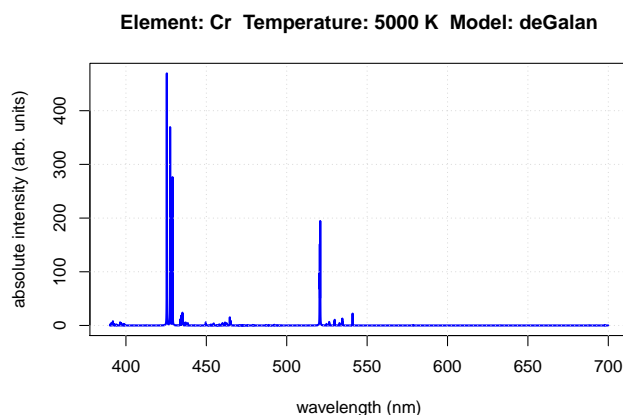
```
plotAES(file, lambda.min = 390, lambda.max = 700,
        signal_type = c("relative", "absolute"),
        main_title = NULL)
```

with its arguments and their default values defined below:

- **file**: The name of an object created using `simulateAES`.
- **lambda.min** and **lambda.max**: The shortest and the longest wavelengths, in nm, that define the limits to display on the *x*-axis. Default values are 390 nm and 700 nm, respectively, but any values between 390 nm and 700 nm are possible.
- **signal\_type**: Indicates how emission intensity is scaled on the *y*-axis; one of `"relative"`, which scales the *y*-axis to a maximum value of 1, or `"absolute"`, which scales the *y*-axis using the actual emission intensities. Defaults to `"relative"`.
- **main\_title**: An optional main title. If `NULL`, then a default title gives the element's atomic symbol, the temperature, and the model used to calculate the temperature-dependent partition function. To suppress the main title, enter `main_title = ""`.

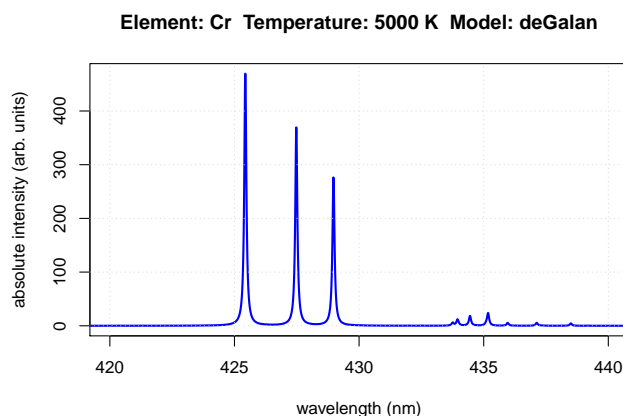
For example, to view the atomic emission spectrum for chromium we enter the following line of code

```
plotAES(file = chromium, signal_type = "absolute")
```



We can examine more closely the cluster of peaks that begins around 425 nm by adjusting the wavelength limits; thus

```
plotAES(file = chromium, signal_type = "absolute",
        lambda.min = 420, lambda.max = 440)
```



## Plotting an Atomic Emission or Atomic Absorbance Line Spectrum

The `plotLines` function is used to view an element's emission or absorbance line spectrum; it takes the form

```
plotLines(file, lambda.min = 390, lambda.max = 700,
          spectrum_type = c("emission", "absorbance"),
          emission_bg = c("white", "black"),
          scale_emission = TRUE,
          main_title = NULL
        )
```

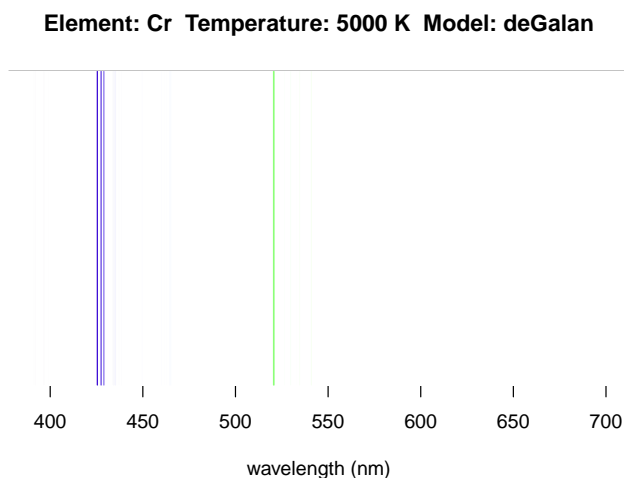
with its arguments and their default values defined below:

- **file**: The name of an object created using `simulateAES`.
- **lambda.min** and **lambda.max**: The shortest and the longest wavelengths, in nm, that define the limits to display on the *x*-axis. Default values are 390 nm and 700 nm, respectively, but any values between 390 nm and 700 nm are possible.
- **spectrum\_type**: The type of line spectrum to display, with options for "emission" or "absorbance". Defaults to "emission".

- **emission\_bg**: The background color on which the emission lines are displayed, with options for "white" or "black". Defaults to "white" and is ignored if **spectrum\_type** is set to display an absorbance line spectrum.
- **scale\_emission**: A logical value that indicates whether the transparency of an emission line is scaled to reflect its relative emission strength. Defaults to TRUE and is ignored if **spectrum\_type** is set to display an absorbance line spectrum.
- **main\_title**: An optional main title. If NULL, then a default title gives the element's atomic symbol, the temperature, and the temperature-dependent partition function. To suppress the main title, enter `main_title = ""`.

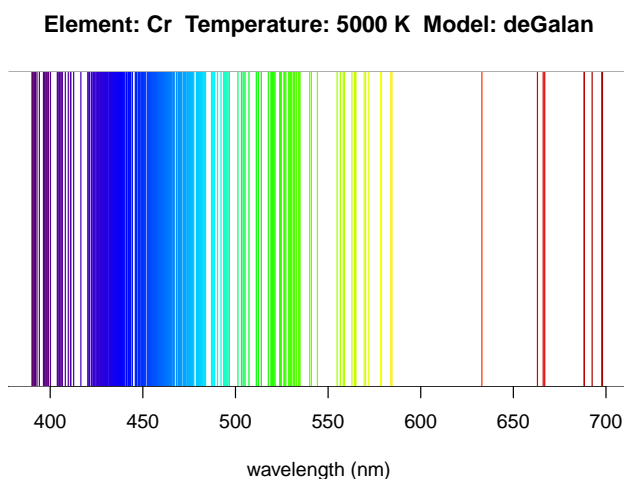
For example, to view the atomic emission line spectrum for chromium, we enter the following code accepting the default values

```
plotLines(file = chromium)
```



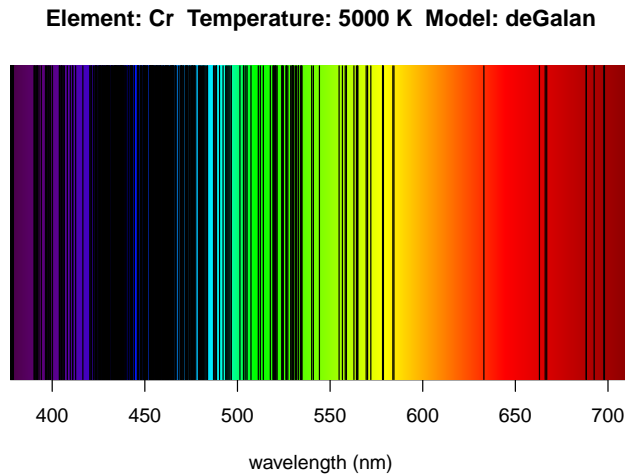
Note that each emission line is displayed with a color that corresponds to its wavelength and that the intensity of the color reflects the line's relative intensity. Because some emission lines are not sufficiently intense, they are difficult (or impossible) to see when `scale_emission` is set to TRUE. To view all emission lines, set `scale_emission` to FALSE

```
plotLines(file = chromium, scale_emission = FALSE)
```



Atomic absorption line spectra are displayed as black lines superimposed on a background that shows the visible spectrum.

```
plotLines(file = chromium, spectrum_type = "absorbance")
```



## Identifying Peaks and Peak Intensities

The `identifyPeaks` function is used to identify peaks with intensities that exceed a user-selected threshold intensity; it takes the form

```
identifyPeaks(file, signal_type = c("relative", "absolute"),  
              percent_max = 1, num_peaks = 5,  
              sortby = c("intensity", "wavelength")  
            )
```

with its arguments and their default values defined below:

- **file**: The name of an object created using `simulateAES`.
- **signal\_type**: Indicates how emission intensity is reported; one of **"relative"**, which reports intensities scaled to a maximum value of 1, or **"absolute"**, which reports the actual emission intensities. Defaults to **"relative"**.
- **percent\_max**: A threshold value for determining which peaks to return, expressed as a percentage of the maximum relative or absolute intensity. Defaults to 1, which returns peaks with intensities greater than 1% of the maximum intensity.
- **num\_peaks**: The maximum number of peaks to report. Defaults to 5 peaks.
- **sortby**: Indicates how the peaks are returned; one of **"intensity"**, which sorts the peaks from greatest-to-smallest intensity or **"wavelength"**, which sorts the peaks from shortest-to-longest wavelength. Defaults to **"intensity"**.

The `identifyPeaks` function makes use of the `findpeaks` function in the `pracma` package, which must be installed. To find the 10 most intense peaks in chromium's atomic emission spectrum, sorted by wavelength, that exceed a threshold of 10% of the absolute intensities, we use the following code, which assigns the output to an object

```
cr_peaks = identifyPeaks(file = chromium, signal_type = "absolute",  
                        num_peaks = 10, percent_max = 10,  
                        sortby = "wavelength")
```

The function returns the peaks as a data frame with two columns, one that gives the wavelengths, in nm, and one that gives the relative or the absolute intensities. To view the results in the console, we can simply enter the object's name

```
cr_peaks
```

```
## Wavelength (nm) Absolute Intensity
## 1 425.44 4.694e+02
## 2 427.48 3.694e+02
## 3 428.97 2.762e+02
## 4 520.45 9.683e+01
## 5 520.60 1.513e+02
## 6 520.84 1.946e+02
```

Alternatively, we can use a function, such as `kable` from the `knitr` package to include the output in a pdf report (as shown below) or in an html file

```
library(knitr)
kable(cr_peaks, align = c("c"), caption = "Most Intense Peaks in Cr Atomic Emission Spectrum")
```

Table 2: Most Intense Peaks in Cr Atomic Emission Spectrum

Wavelength (nm)	Absolute Intensity
425.44	4.694e+02
427.48	3.694e+02
428.97	2.762e+02
520.45	9.683e+01
520.60	1.513e+02
520.84	1.946e+02

Note that although we set `num_peaks` to 10, only six peaks exceed the threshold of 10%.

## Computational Details

The simulations in this Shiny App use the computational approach outlined in Flannigan, D. J. “Spreadsheet-Based Program for Simulating Atomic Emission Spectra” *J. Chem. Educ.*, **2014**, *91*, 1736–1738. The temperature-dependent peak intensities,  $S$ , for the transition of an electron from a higher energy state to a lower energy state are calculated using the following equation

$$S = \frac{g_k A_{ki}}{Q(t) \lambda_c} e^{-E_k/k_B T}$$

where  $E_k$  is the energy, in eV, of the higher energy state,  $A_{ki}$  is the transition probability between the higher energy state and the lower energy state,  $g_k$  is the statistical weight of the higher energy state (that is, the number of quantum states with the same energy),  $\lambda_c$  is the wavelength at the center of the emission peak,  $k_B$  is Boltzmann’s constant,  $T$  is the temperature in K, and  $Q(t)$  is a temperature-dependent partition function. Values for  $\lambda_c$ ,  $E_k$ ,  $A_{ki}$ , and  $g_k$  are obtained from the NIST Atomic Spectra Database (Kramida, A., Ralchenko, Yu., Reader, J. and NIST ASD Team (2018). NIST Atomic Spectra Database (version 5.5.6), [Online]. Available: <https://physics.nist.gov/asd>. National Institute of Standards and Technology, Gaithersburg, MD.) and used here with permission.

Values for the temperature-dependent partition function,  $Q(t)$ , are estimated here using an empirical model developed by Irwin (see Irwin, A. W. “Polynomial Partition Function Approximations of 344 Atomic and Molecular Species” *The Astrophysical Journal Supplement Series*, **1981**, *45*, 621–633), which is valid for temperatures between 1000 K and 16000 K. Irwin’s model fits the following equation

$$\ln Q = \sum_{i=0}^5 a_i (\ln T)$$

to published partition coefficients where the six  $a_i$  values are the model's adjustable fitting parameters.

The calculated peak intensity is then used to construct a Lorentzian line shape where intensity,  $I$ , is expressed as a function of wavelengths from 390 nm to 700 nm.

$$I = \frac{2S}{\pi} \times \frac{w}{4(\lambda - \lambda_c)^2 + w^2}$$

where  $w$  is the peak's width. A summation at each wavelength of the values of  $I$  for each peak yields the overall emission spectrum.

Irwin's model is one of several for calculating partition coefficients as a function of temperature. The paper by Flannigan, noted above, uses the deGalan model (deGalan, L.; Smith, R.; Winefordner, J. D. "The Electronic Partition Function of Atoms and Ions Between 1500 K and 7000 K", *Spectrochim. Acta*, **1968**, *23B*, 521-525). Another option is the Tamaki model (Tamaki, S.; Kuboda, T. "The Electronic Partition Function of Atoms and Ions Between 7000 and 12000 K", *Spectrochim. Acta*, **1987**, *42B*, 1105-1111). As shown here for chromium

all three models provide a reasonable fit to the published partition coefficients for their temperature limits. The Irwin model is used here because it covers a wider range of temperatures.

## Resources

### element: chromium

