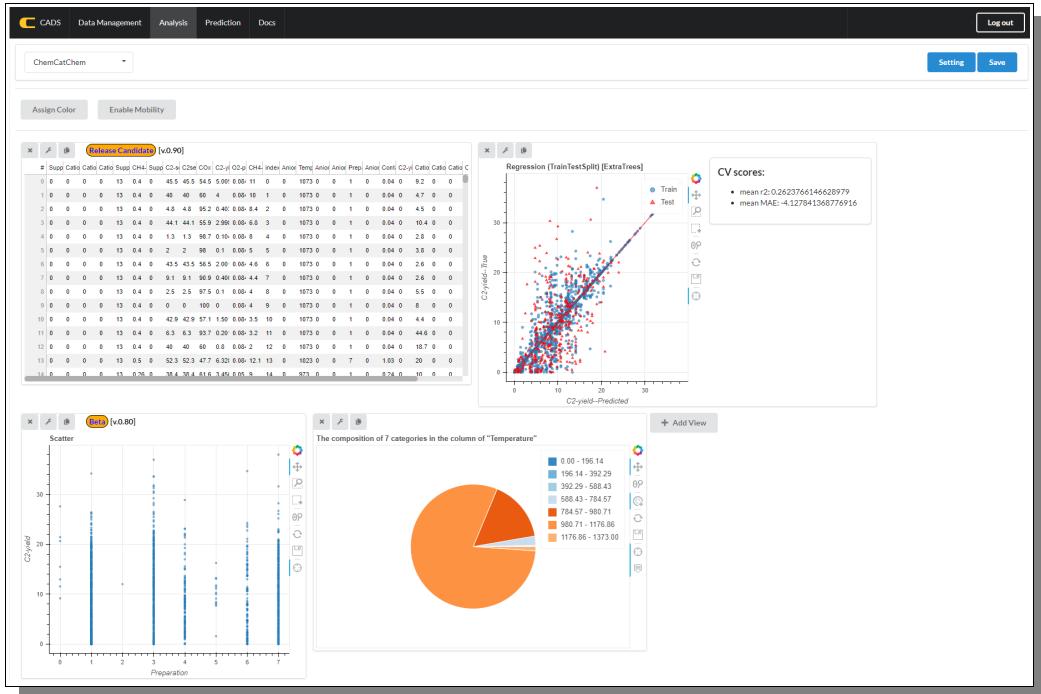
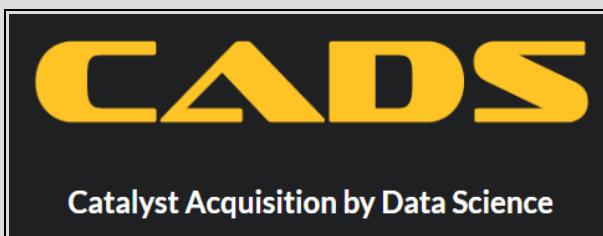


# TAKAHASHI GROUP (情報化学研究室)

Hokkaido University



**CATALYST ACQUISITION BY DATA SCIENCE**



**CADS**  
Catalyst Acquisition by Data Science



**CADS**

# Components Manual

ENGLISH

# Table of Contents

Table of Contents.....	1
Preface.....	2
Introduction.....	3
CADS Overview.....	3
Data Management.....	3
Analysis.....	3
Prediction.....	3
CADS Architecture.....	4
CADS Online in the Browser.....	4
Components Manual.....	5
Shared Features.....	5
Connections.....	6
Visualization Components.....	6
Table.....	6
Scatter.....	8
Bar.....	10
Pie.....	13
Scatter 3D.....	14
Line.....	16
Data Processing Components.....	18
ImageView.....	18
Analysis Components.....	27
Parallel Coordinates.....	27
Histogram.....	29
Feature Importance (FD).....	30
Heat map.....	31
Pairwise Correlation.....	32
NodeGraph.....	33
Machine Learning Components.....	40
Regression.....	40
Classification.....	43
K-Means Clustering.....	46
Gaussian Process.....	49
Statistics.....	52
Static Data Visual Support Components.....	54
Periodic Table.....	54
Molecule 3D.....	55
Summary.....	59
Index.....	60
Figure Index.....	62

# Preface

*A few words from the creators.*

This manual is attempting to be an extensive guide to anyone who wishes to use the CADS web application found online at <https://cads.eng.hokudai.ac.jp> in specific regard to Analysis components and how to extensively use them

CADS stands for 'Catalyst Acquisition by Data Science' and is an Integrated Web-based Visual Platform designed for catalytic data analysis for catalyst developers and researchers in Materials Informatics. The platform is aiming to help scientists (mainly in the material science field but not exclusively) design new materials with the sharing of material data and analysis tools and support for trial-and-error process of data analysis with user-friendly interactive visual interfaces.

In short; CADS is basically a visualization and analyzing tool.

The platform basically contains of three parts; 'Data Management' where one can upload data sources for data analysis and also access publicly shared data sources, 'Analysis' where one can analyze and visualize data sources in an exploratory visual analytics environment using a wide range of customizable components in personal workspace, and 'Prediction' where one can predict materials properties with stored pre-trained models.

This document aims to help users understand in details what each, currently available, analysis-component can do, and how and why.

For knowledge on understanding CADS in a more general sense, please refer to the proper user manual.

CADS is a free platform that does not require additional software installation. Site operation has been tested through Google Chrome, Mozilla Firefox, and Microsoft Edge. Site operation has also been tested on Windows, Mac, and Linux operating systems. Should bugs or other issues arise using the platform, please do not hesitate to contact us. We welcome any and all feedback for bugs, site usage, and additional functions for data analysis for improving this platform.

CADS is still a work in progress and our goal is to make it user friendly and helpful for all data scientist out there.

This work is funded by Japan Science and Technology Agency(JST) CREST, JSPS KAKENHI Grant-in-Aid for Young Scientists and Materials research by Information Integration Initiative project of the Support Program for Starting Up Innovation Hub from JST.

To view or access the CADS code, go to GitHub here: <https://github.com/Material-MADS/mads-app>

September 21, 2023

*CADS Software System Developer*

Micke Nicander Kuwahara

# Introduction

*Get to know what CADS is and what it can do for you as a researcher.*

CADS is a catalyst analysis environment that implements a user-friendly graphic user interface(GUI) that aims to help scientist to analyze data in a more simple and effective way without the need of coding and/or setting up complex IT environments.

## CADS Overview

CADS provides three basic functions for users: “Data Management”, “Analysis”, and “Prediction”.

### Data Management

CADS provides the ability to upload and record individual data as well as data analysis procedures with the option for multi-user sessions, allowing users to share data and analysis reports with collaborators of their choosing. Meaning that not only can a researcher study and analyze their own data, but they can also compare and study other's shared data for a better and deeper understanding.

### Analysis

Data visualization tools (scatter plotting, histograms, etc.) and machine learning functions are available for catalyst data analysis and machine learning model selection. This is the core part of the CADS platform and one of its major strengths. In the workspace area a user can load a various buffet of components through which the user can then analyze and manipulate the data for make new findings and getting further understanding. The range of Components is steadily growing and the power of this section is increasing every month. Besides the classic visualization tools mentioned above, there are many more, ranging from image manipulation, machine learning and advanced analyzing and visualization tools, all with simple and straight forward user interfaces that guides the user to do complex analyzing with few and user-friendly operations.

This manual's primary target is to describe and explain how to use each of the currently available Components that CADS contain.

### Prediction

Data prediction capabilities are available within the “Analysis” function and inside the Machine Learning components, allowing users to apply and store learning models for predicting new output (e.g. predicting new catalysts). Under the Prediction tab the user can then apply those pre-trained and saved models in order to predict the result with new data.

## CADS Component Manual

### CADS Architecture

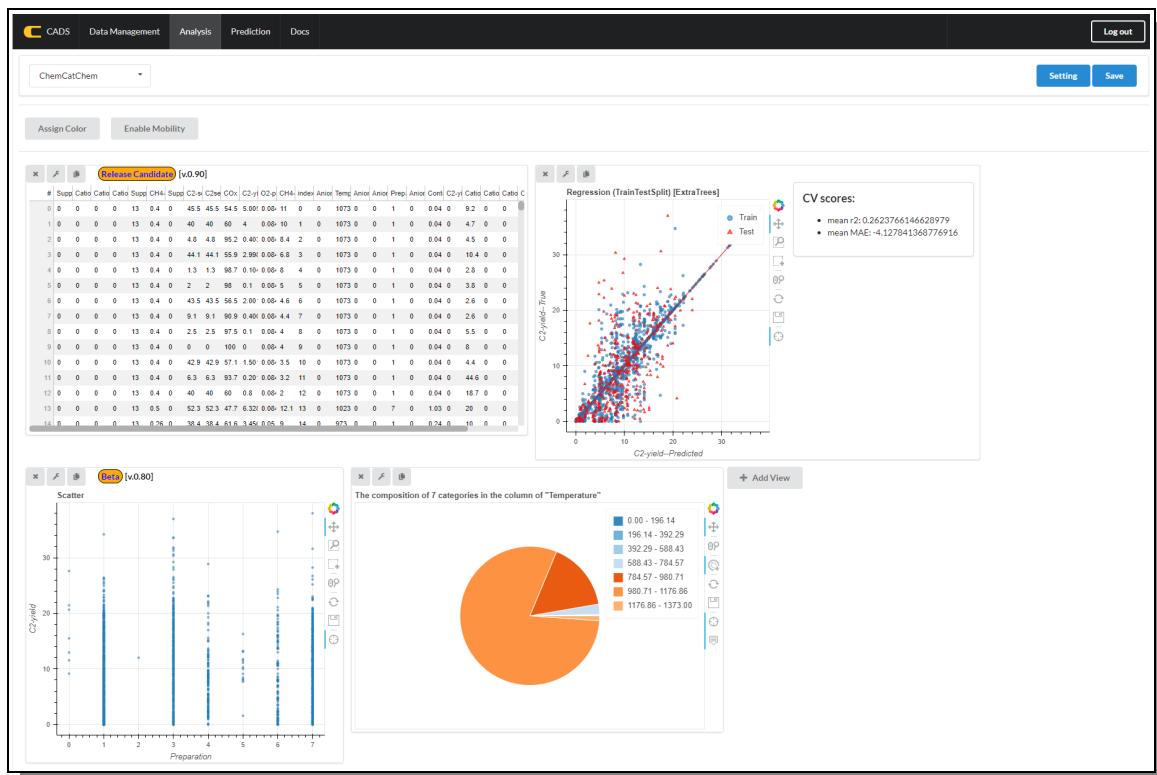
On the surface to the average user the CADS system is just an ordinary website operated by basic mouse and keyboard operations. A few clicks and perhaps some simple typing of values and then submit to the CADS server, and voila, the data will be displayed in a new graphical and exciting way.

Under the hood the CADS system is built with Django and React JS and for easy maintenance and sound and familiar structure. The custom and generic Analyzing components are using state of the art technology from various high-end libraries for graphical display of data, e.g. Bokeh JS and Plotly JS. On the server side a wide range of support system is feeding the components with all sorts of data management capabilities like SciKit Learn and SciKit Image and much more. This is not needed to be known for the CADS user, but it might be good to know that under the hood are all the well-known complexity the research world is used to, but through CADS, most of the complexity is removed or toned down. Data analyzing should not just be for IT savvy people, it should be available to everyone.

### CADS Online in the Browser

In order to use CADS to its fullest one need to have a user account in order to be able to save data and workspaces. Applying for an account is simple and quick and after confirming with a valid email address the CADS system is fully available to use. All interaction from hereon is based on familiar browser interactions using classic form fields and buttons and no need for any plugins or additional downloads are required. All work can be saved and if wanted to, relatively easily shared.

For more details on CADS in general terms, please look for the CADS user Manual. Here we will now dive deep into the world of CADS components and explain what they are and what the do and how one use them. So, let's go into those details in the next chapter.



# Components Manual

CADS contains a wide variety of different data analyzing components and in this chapter we will look at each and everyone of those components in details.

**C**ADS components are available in the Analysis section of CADS which is described in detail in the User Manual together with all general parts of the CADS application and will not be repeated here. So if you are not yet familiar with CADS at all in general or the Analysis Section in particular we recommend that you first read the above mentioned User Manual found on the CADS website.

In this manual we will immediately dive into the sea of available components and explain them one by one in an orderly fashion so that you can use them properly and make most out of them.

Components are divided into five different groups for more easy localization, but these groups are not exclusive and some components would probably fit in more than one groups, but for the purpose of organization we divide them as such.

## Shared Features

All CADS components tries to look familiar to each other in some sense so that users can more easily identify features they are looking for, so interactions and forms should not look too different between different component, even if there are of course differences. Most components are build using the Bokeh library, meaning it shares some style and behavior and below we will point out those specific Bokeh recognizable features.



### Bokeh Supported Component's Buttons

- 
- 1: **Link**, This takes you to the Bokeh website for all information on the library if ever needed.
  - 2: **Pan**, Let you move the graph/chart inside the Bokeh box in horizontal and vertical directions.
  - 3: **Box Zoom**, let you zoom into an area that you select via box drawing.
  - 4: **Box Select**, let you select parts of the chart (and usually highlight it) via box drawing.
  - 5: **Wheel Zoom**, When enabled let you zoom the chart using the mouse wheel.
  - 6: **Reset**, This resets the chart to look like it did from the beginning before the user moved and zoomed etc.

## CADS Component Manual

- 7: **Save**, Let the user save an image of the chart as it currently looks.
- 8: **Crosshair**, when enabled the user can see exactly where the mouse pointer is by a crosshair being displayed inside the graph.
- 9: **Tap**, when enabled it lets the user tap select a data area to highlight it.
- 10: **Hover**, when enabled the user can get information on the current mouse hovered over data points displayed in a popup message box.

CADS also have other components using different libraries, when Bokeh is not available for a specific goal, e.g. Plotly, VivaGraph or Vanilla JS. Those are mainly used for only one or perhaps two specific components though so their features will be explained in that section of the manual.

## Connections

All loaded components share the same data, but they also, in cases where that makes sense, share selection status. So if one would select some portion of the data in one specific loaded component, then that selection might be automatically visible also in others for a helpful way to identify data points even though visualized differently in many various components.

## Visualization Components

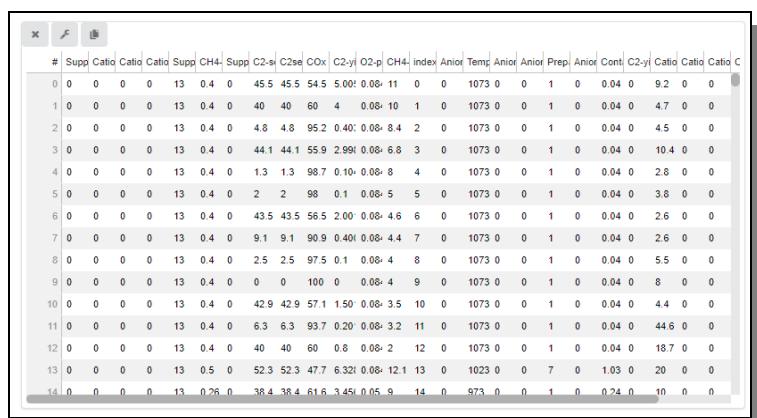
The visualization components are those that mainly aim to visualize data in classical ways that most statistician are used to. It seldom or never change or alter or enhance the data itself, it just display it in a suitable graphical way. Classical charts such as Pie Charts and Bar charts is included in this category.

## Table

The Table component is as simple and as straightforward as it gets.

In the same way as one can study the data inside the Data Management section one can load it also in the visualization section into a table component. If data is already available in the workspace, then the Table component will automatically use that data and display it accordingly. In Figure 1 one can see such a component loaded with data attached to it.

The table will draw itself as visually pleasing as possible, given its outer boundaries which can be changed inside the component form. In Figure 2 we see the component form with its available fields. Which are also listed below with some further description.



A screenshot of a classic table component. The table has a header row with various column names: #, Supp, Catio, Calio, Catio, Supp, CH4, Supp, C2-si, C2se, COX, C2-yi, O2-p, CH4, index, Anior, Temp, Anior, Anior, Prep, Anior, Conti, C2-yi, Catio, Calio, C. The body of the table contains 15 rows of data, each with numerical values corresponding to the headers. The table is presented in a standard grid format with horizontal and vertical lines separating the cells.

#	Supp	Catio	Calio	Catio	Supp	CH4	Supp	C2-si	C2se	COX	C2-yi	O2-p	CH4	index	Anior	Temp	Anior	Anior	Prep	Anior	Conti	C2-yi	Catio	Calio	C
0	0	0	0	0	13	0.4	0	45.5	45.5	54.5	5.00:	0.08-	11	0	0	1073	0	0	1	0	0.04	0	9.2	0	0
1	0	0	0	0	13	0.4	0	40	40	60	4	0.08-	10	1	0	1073	0	0	1	0	0.04	0	4.7	0	0
2	0	0	0	0	13	0.4	0	4.8	4.8	95.2	0.40:	0.08-	8.4	2	0	1073	0	0	1	0	0.04	0	4.5	0	0
3	0	0	0	0	13	0.4	0	44.1	44.1	55.9	2.99:	0.08-	6.8	3	0	1073	0	0	1	0	0.04	0	10.4	0	0
4	0	0	0	0	13	0.4	0	1.3	1.3	98.7	0.10:	0.08-	8	4	0	1073	0	0	1	0	0.04	0	2.8	0	0
5	0	0	0	0	13	0.4	0	2	2	98	0.1	0.08-	5	5	0	1073	0	0	1	0	0.04	0	3.8	0	0
6	0	0	0	0	13	0.4	0	43.5	43.5	56.5	2.00:	0.08-	4.6	6	0	1073	0	0	1	0	0.04	0	2.6	0	0
7	0	0	0	0	13	0.4	0	9.1	9.1	90.9	0.40:	0.08-	4.4	7	0	1073	0	0	1	0	0.04	0	2.6	0	0
8	0	0	0	0	13	0.4	0	2.5	2.5	97.5	0.1	0.08-	4	8	0	1073	0	0	1	0	0.04	0	5.5	0	0
9	0	0	0	0	13	0.4	0	0	0	100	0	0.08-	4	9	0	1073	0	0	1	0	0.04	0	8	0	0
10	0	0	0	0	13	0.4	0	42.9	42.9	57.1	1.50:	0.08-	3.5	10	0	1073	0	0	1	0	0.04	0	4.4	0	0
11	0	0	0	0	13	0.4	0	6.3	6.3	93.7	0.20:	0.08-	3.2	11	0	1073	0	0	1	0	0.04	0	44.6	0	0
12	0	0	0	0	13	0.4	0	40	40	60	0.8	0.08-	2	12	0	1073	0	0	1	0	0.04	0	18.7	0	0
13	0	0	0	0	13	0.5	0	52.3	52.3	47.7	6.32:	0.08-	12.1	13	0	1023	0	0	7	0	1.03	0	20	0	0
14	0	0	0	0	13	0.26	0	38.4	38.4	61.6	3.45:	0.05	9	14	0	973	0	0	1	0	0.24	0	10	0	0

Figure 1: A classic table showing the data in rows and columns

### Table Component's Form Fields

## CADS Component Manual

**Filter:** If rows are selected and color coded, then by using filter one can hide all other data from view.

**Columns:** If none is selected, all columns show, but if some are selected only those that are shows.

**Extent:** All components have width and height values encompassed in the extent group.

If the form field 'filter' is empty, it just means that no colors have been assigned in the workspace and no rows have been properly selected, which is done outside the component form.

The screenshot shows the 'Table [1]' dialog box. It contains three main sections: 'Filter' (with a dropdown menu showing 'ColorTags'), 'Columns' (with a dropdown menu showing 'Columns'), and 'Extent' (with input fields for width '800' and height '400'). At the bottom right are 'Cancel' and 'Submit' buttons.

Figure 2: The Table Form is as simple as it gets.

In Figure 3 we see how the table component looks when just a few columns are selected and also some of the rows have been selected and assigned a specific color. Default select color is orange.

#	Preparation	Temperature	CH4-pressure	C2-yield
0	1	1073	0.4	5.005
1	1	1073	0.4	4
2	1	1073	0.4	0.4032
3	1	1073	0.4	2.9988
4	1	1073	0.4	0.104
5	1	1073	0.4	0.1
6	1	1073	0.4	2.001
7	1	1073	0.4	0.4004
8	1	1073	0.4	0.1
9	1	1073	0.4	0
10	1	1073	0.4	1.5015
11	1	1073	0.4	0.2016
12	1	1073	0.4	0.8
13	7	1023	0.5	6.3283
14	1	973	0.26	3.456

Figure 3: A formatted table with selected columns and selected rows

In Figure 4 we see how the form looks like for the configuration setup.

## CADS Component Manual



Figure 4: One can filter which rows to be seen and select which rows to be visible.

And in Figure 5 below how it is being visualized with only a few rows are shown.

The screenshot shows a modified Table component. The header row has columns for "#", "Preparation", "Temperature", "CH4-pressure", and "C2-yield". Below are four data rows, each with a "#": 4, 5, 6, and 7. The "Preparation" column for all rows contains the value "1". The "Temperature" column for all rows contains the value "1073". The "CH4-pressure" column for all rows contains the value "0.4". The "C2-yield" column for all rows contains the value "0.104", except for row 6 which is "0.1" and row 7 which is "0.4004". The entire table is enclosed in a light gray border.

#	Preparation	Temperature	CH4-pressure	C2-yield
4	1	1073	0.4	0.104
5	1	1073	0.4	0.1
6	1	1073	0.4	2.001
7	1	1073	0.4	0.4004

Figure 5: A modified Table component for a more specific look and feel.

## Scatter

The classic scatter plot component looks like shown in Figure 6 with its familiar Bokeh buttons explained earlier to the right and a typical Scatter plot look.

In Figure 7 one can see the Scatter Plot component form with its configurable fields, explained more in details below.

### Scatter Component's Form Fields

**Filter:** If rows are selected and color coded, then by using filter one can hide all other data from view.

**X:** Which data column to show on the C-axis.

**Y:** Which data column to show on the Y-axis.

**Color Assignment:** If enabled it can use a color range to give each data point a color based on a specific column value.

**Color:** Which column to use for selecting colors

**Extent:** All components have width and height values encompassed in the extent group.

## CADS Component Manual

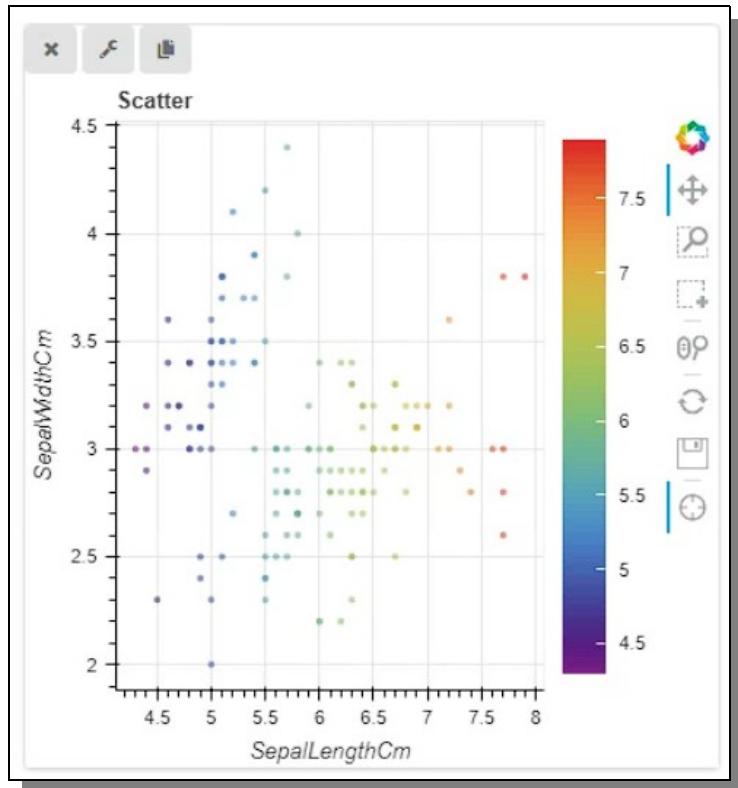


Figure 6: The Scatter plot components displaying Iris data

### Scatter [2]

Filter: ColorTags

X: SepalLengthCm

Y: SepalWidthCm

Color assignment:

Color: SepalLengthCm

Extent: 400 400

**Cancel** **Submit**

Figure 7: The Scatter plot form configured for a particular view

## CADS Component Manual

### Bar

The classic Bar Chart component looks like shown in Figure 8 with its familiar Bokeh buttons explained earlier to the right and a typical Bar Chart look.

In Figure 9 and Figure 11 one can see the Bar Chart component form with its configurable fields, explained more in details below.

#### *Bar Component's Form Fields*

---

**Enable Transpose View:** If enabled it shows more fields that let the user easily transpose the data for new angles to view it.

**Transpose Group Members:** Select the columns which should form a new group in the transposed view.

**Transpose Group Label:** Give the transpose group a name that allows us to use it.

**Transpose Split Column:** Which column that we wish to split up into its own range of columns.

**Dimensions:** The categorical unique values displayed on the x-axis (if not unique, the values will be needed to calculated so that they can be treated as categorical unique in some way)

**Value Calculation Method:** If the Dimension column does not contain purely unique data points in every single row, then this algorithm needs to be applied to make sure all points are unique. The following methods are available:

- Total
- Mean
- Median
- Max
- Min

**Measures:** One or many columns containing numerical values for the specified Dimension used on the Y-axis.

**Filter:** If data are selected and color coded, then by using filter one can hide all other data from view.

**Extent:** All components have width and height values encompassed in the extent group.

**Bar Directions:** If the bars should be drawn in vertical or horizontal direction.

**Legend Location:** Where the chart legend should be displayed for best view.

**Color Palette:** Which palette of colors that should be used when drawing multiple bars per category.

Anyone familiar with a Bar chart should have no problem configuring the form to get something to show. If the component detects that the x-axis values are not unique as expected, it will tell the user to apply some algorithm so that it can be 'grouped' in a unique way, e.g. Mean or Median.

What is perhaps slightly unique with the CADS Bar Component though is the easy capability to transpose the data before displaying it.

In Figure 10 we see the same data used in Figure 9 but transposed differently. Figure 11 Shows the setting in the form for creating a transposed view with only a few fields.

## CADS Component Manual

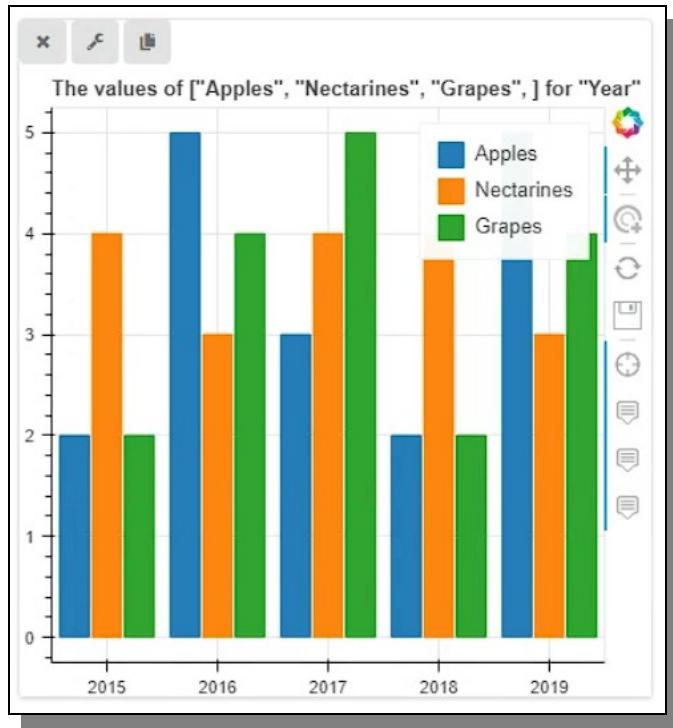


Figure 8: A classic bar chart showing multiple data points.

The figure shows the configuration interface for a Bar chart, titled "Bar [1]". The interface includes the following settings:

- Enable Transpose View**: A toggle switch that is off.
- Dimension**: Set to "Year".
- Measures**: Set to "Apples", "Nectarines", and "Grapes".
- Filter**: Set to "ColorTags".
- Extent**: Width 528, Height 350.
- Bar Direction**: Set to "Vertical".
- Legend Location**: Set to "top\_right".
- Color Palette**: Set to "Category10 (max 10 colors)".
- Buttons**: "Cancel" and "Submit" buttons at the bottom right.

Figure 9: The Bar Chart form default view configured to display the data in shown specific way.

## CADS Component Manual

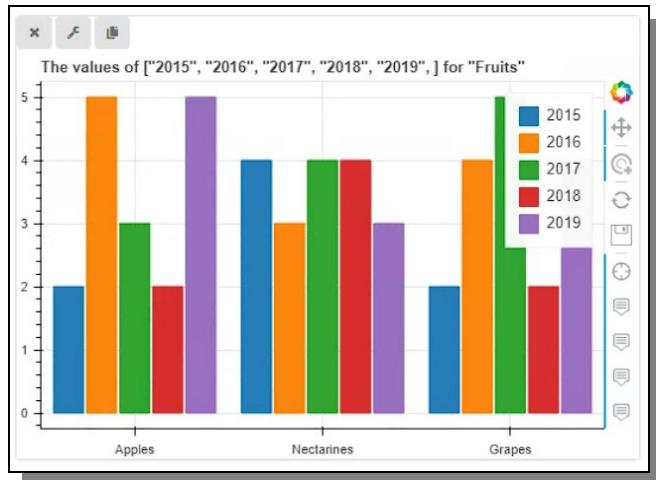


Figure 10: Transposed data displayed in the Bar Chart component.

Bar [1]

Enable Transpose View ⓘ:

Transpose Group Members ⓘ:

Transpose Group Label ⓘ:

Transpose Split Column ⓘ:

Dimension ⓘ:

Measures ⓘ:

Filter:

Extent: Width: 528 Height: 350

Bar Direction:

Legend Location:

Color Palette (if number of bins exceed number of colors available in the palette, default palette will be used):

(Max Colors: 10)

Figure 11: The Bar component easily lets the user transpose the data for viewing in new angles.

## CADS Component Manual

### Pie

The classic Pie Chart component looks like shown in Figure 12 with its familiar Bokeh buttons explained earlier to the right and a typical Pie Chart look.

In Figure 13 one can see the Pie Chart component form with its configurable fields, explained more in details below.

#### *Pie Component's Form Fields*

**Target Column:** The data column that should be used in the pie chart.

**Number of Bins:** The amount of slices the pie should be divided into.

**Include Undefined Values:** If enabled the blank and null values will also be included in the drawing of the pie chart, otherwise ignored.

**Extent:** All components have width and height values encompassed in the extent group.

**Color Palette:** Which palette of colors that should be used when drawing multiple slices per category.

Most of the Pie chart is automatic and should therefore be very easy to use.

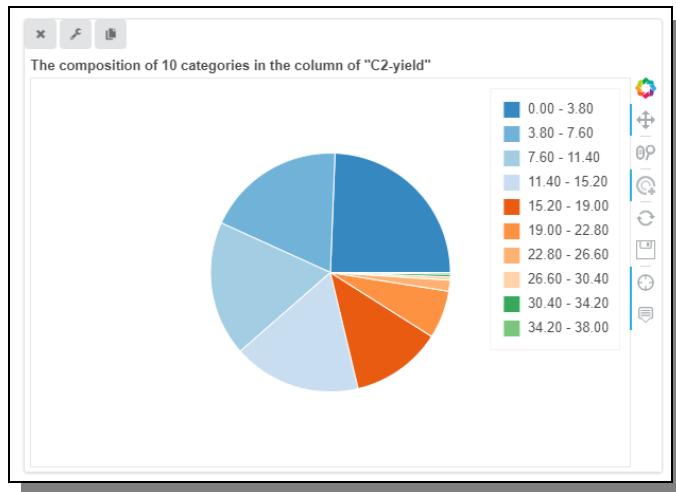


Figure 12: A classic Pie Chart displaying its data

A screenshot of the Pie Chart component configuration form. The title bar says "Pie [1]". The form includes fields for "Target column" (set to "C2-yield"), "Number of bins" (set to 10), and "Include undefined values" (unchecked). It has sections for "Extent" (Width 0, Height 0) and "Color Palette" (set to "Category20c (max 20 colors)"). At the bottom are "Cancel" and "Submit" buttons.

Figure 13: The Pie Chart Component configuration Form for easy and quick views

## Scatter 3D

The Scatter 3D component let the user display data in three dimensions and interact with it in the graph view easily using the mouse to rotate, pan and zoom around data points. In Figure 14 below one can see how the component looks like with data inside.

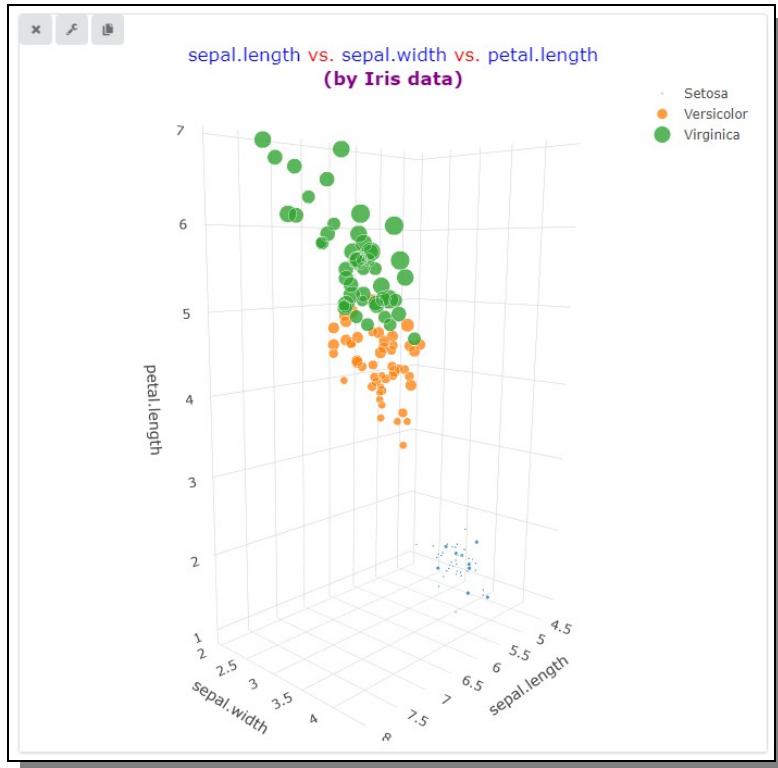


Figure 14: The Scatter 3D allows us to interact with the data in three dimensions.

As seen in Figure 15 below the Scatter 3D configuration form contains an easy and small set of fields for controlling the view. These fields are explained more in detail in the list below.

### Scatter 3D Component's Form Fields

**Method:** If set to Manual then the user just assign one data column to each axis (x, y and z). But if set to PCA (Principal component analysis) then the data will be pre-calculated and shown for PCA display instead.

**X:** (In Manual Method Mode), The column that should be used on the X-axis.

**Y:** (In Manual Method Mode), The column that should be used on the Y-axis.

**Z:** (In Manual Method Mode), The column that should be used on the Z-axis.

**Feature columns:** (In PCA Method Mode), Which columns that should be included in the PCA calculation.

**Target column:** (In PCA Method Mode), The target column the PCA calculation is used upon.

**Color Assignment:** If enabled it allows the user to assign colors to the data points based on a specific column.

**Color Mapping Column:** The column whose values be used to assign colors to data points.

**Color Palette:** Which palette of colors that should be used when drawing multiple slices per category.

**Size Assignment:** If enabled it allows the user to assign size to the data points based on a specific column.

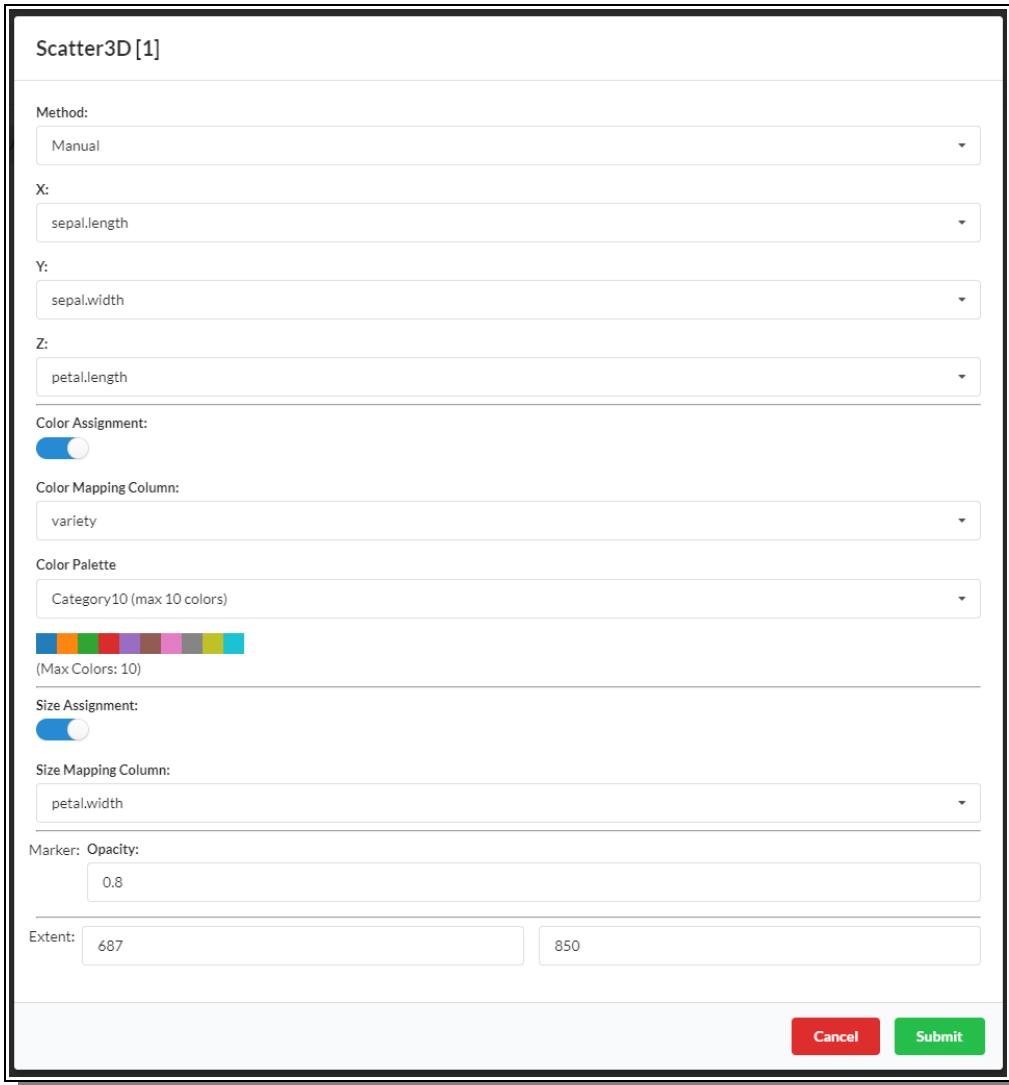
**Size Mapping Column:** The column whose values be used to assign size to data points.

## CADS Component Manual

**Marker Opacity:** The opacity (transparency) value for the data point markers.

**Extent:** All components have width and height values encompassed in the extent group

Using PCA will require some server side calculations that might take a few extra seconds depending of the size of the data, but it should not be of any bigger concern.



The image shows a configuration form for a Scatter3D component. The form is titled "Scatter3D [1]" and contains the following fields:

- Method:** A dropdown menu set to "Manual".
- X:** A dropdown menu set to "sepal.length".
- Y:** A dropdown menu set to "sepal.width".
- Z:** A dropdown menu set to "petal.length".
- Color Assignment:** A toggle switch that is turned on.
- Color Mapping Column:** A dropdown menu set to "variety".
- Color Palette:** A dropdown menu set to "Category10 (max 10 colors)". Below it is a color swatch bar showing ten colored squares, labeled "(Max Colors: 10)".
- Size Assignment:** A toggle switch that is turned on.
- Size Mapping Column:** A dropdown menu set to "petal.width".
- Marker: Opacity:** A text input field containing "0.8".
- Extent:** Two text input fields showing "687" and "850".

At the bottom right are two buttons: "Cancel" (red) and "Submit" (green).

Figure 15: The Scatter 3D configuration form lets us setup the view in a quick and easy way

Since the Bokeh library does not support 3D, this component is using the Plotly library instead, meaning it does not have the Bokeh buttons but instead a set of similar Plotly buttons when hovering above the plot graph, as seen in Figure 16 below and is explained in detail in the following list.



Figure 16: The Scatter 3D component uses Plotly and therefore have a slightly different set of interaction buttons attached to it.

### *Plotly Supported Component's Buttons*

- 1: **Save Image**, Download current view of the plot as a JPG image.
- 2: **Zoom**, let you zoom using the mouse buttons, can also be done with the mouse wheel.
- 3: **Pan**, Let you grab and drag the graph/chart inside the Plotly box in horizontal and vertical directions.
- 4: **Orbital rotation**, let you turn around the plot in all directions centered at the middle of the plot.
- 5: **Turntable rotation**, Let you turn the data around in a more controlled form.
- 6: **Reset**, This resets the chart to look like it did from the beginning before the user moved and zoomed etc.
- 7: **Return to last save**, resets the camera to the last save position.
- 8: **Plotly**, take the user to the Plotly website for more information about the library.

## Line

The Line Component let you easily display data in the classic Line Chart way as can be seen in the Figure 17 using a range of configuration possibilities for controlling the look and feel and also showing multiple lines in the same chart.

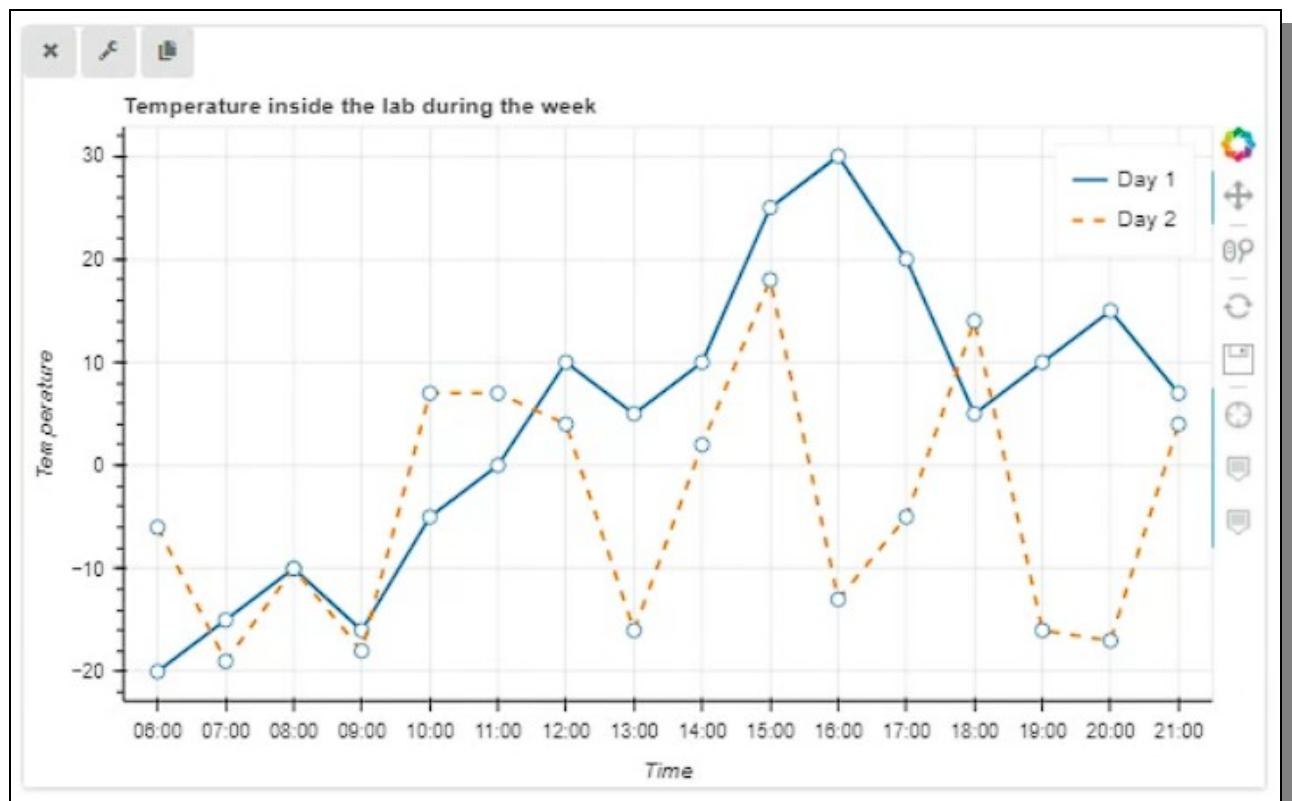


Figure 17: A classic Line Chart for all needs that requires it.

As seen in Figure 18 below it is very straight forward and easy to configure the Line chart inside the Line Configuration form and assign data columns to it as well as make it look and feel the way most suitable for the data and readability.

## CADS Component Manual

The screenshot shows the configuration interface for a 'Line [1]' component. The form is divided into several sections:

- X-Axis:** A dropdown menu set to 'Time'.
- Y-Axis ①:** A dropdown menu containing 'Temp Day 1' and 'Temp Day 2'.
- Title ①:** A text input field containing 'Temperature inside the lab during the week'.
- Labels:** XAxis ①: Time; Y Axis ①: Temperature; Legend ①: Day.
- Extent:** Width: 700; Height: 400.
- Color Palette:** Category10 (max 10 colors). A color palette bar showing ten distinct colors: blue, orange, green, red, purple, brown, pink, yellow, grey, and cyan. Below it, it says '(Max Colors: 10)'.
- Styles:** Enable Different Line Styles ①: A toggle switch that is turned on. Enable Data point Markers ①: A toggle switch that is turned on.
- Buttons:** 'Cancel' (red) and 'Submit' (green).

Figure 18: It could probably not be any easier than this to configure the Line chart component.

The fields available for the Line component and which are found in the Line configuration form are explained more in detail in the list below.

### Line Component's Form Fields

**X-Axis:** Select which data column should be represented by the horizontal axis.

**Y-Axis:** Select which data column(s) should be drawn in the chart as lines and what data type is represented by the vertical axis.

**Title:** The title which represents the chart may be partly auto generated if left blank, but for higher control and understanding, it should probably be manually written.

**Labels - X-Axis:** The X-Axis Label will be auto generated from the data if left blank, but for higher control and understanding, it may be manually written.

**Labels - Y-Axis:** The Y-Axis Label will be auto generated from the data if left blank, but for higher control and understanding, it may be manually written.

**Labels - Legend:** The Legend Label will be auto generated from the data if left blank, but for higher control and understanding, it may be manually written. If multiple Lines are drawn, each line will be numbered.

**Extent:** All components have width and height values encompassed in the extent group

**Color Palette:** Which palette of colors that should be used when drawing lines.

**Enable Different Line Styles:** If enabled each line will be drawn in different style e.g solid, dashed, dotted etc.

**Enable Data Point Markers:** If enabled each data point will be marked by a small circle.

## Data Processing Components

The Data Processing components are those that in some way also tweaks and alter and enhance the data before visualizing it in a new and useful way. CADS are using SciKit Learn (and other useful Python data tools and libraries) under the hood on the server side, and many of this and later category's components uses those features in various ways. In this category there is only one component yet and that is an image manipulator component.

### ImageView

The ImageView component is a very powerful and useful component that let the user load basically any image and then not only display it in many possible ways, but also manipulate and alter it for very many uses, not least scientifically. When the ImageView component is first loaded it contains no image and will then properly show that as seen in the Figure 19 below.

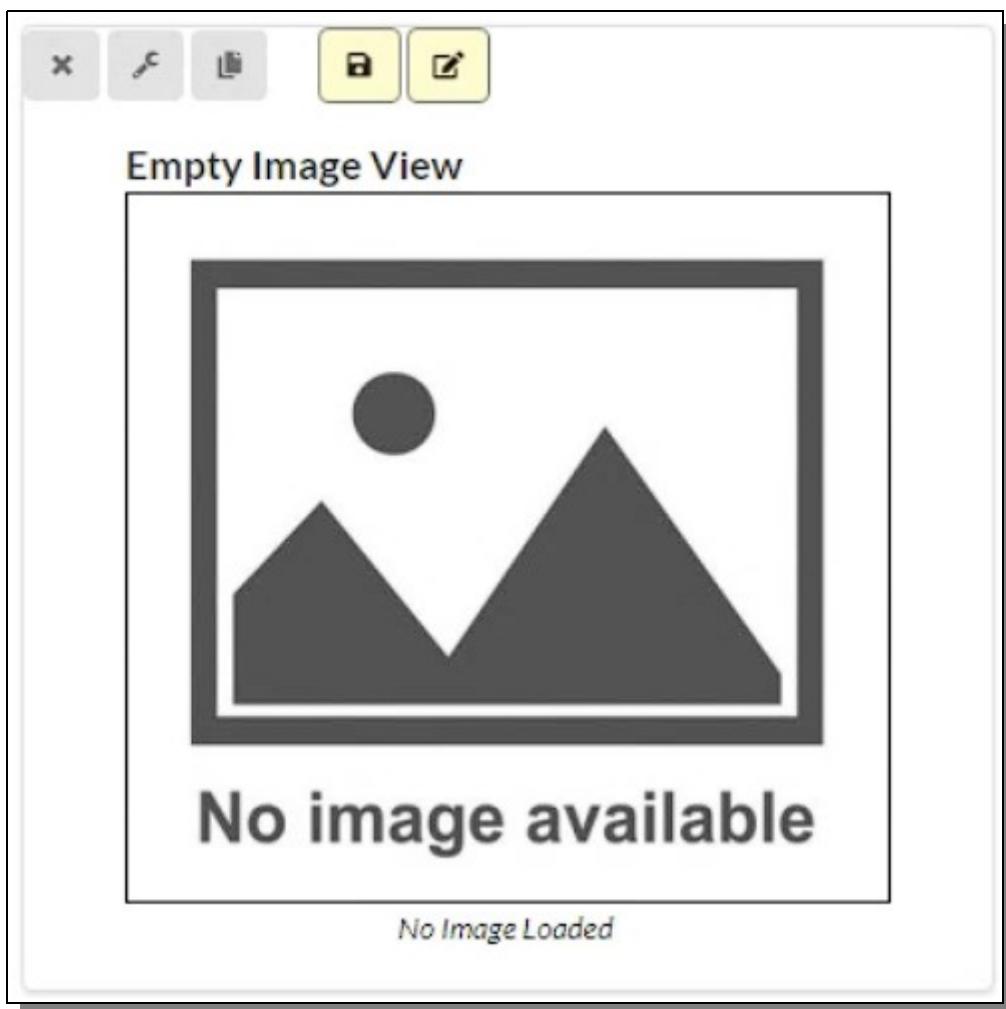


Figure 19: An empty ImageView component that has just been loaded into the workspace.

This component also contains two more additional custom buttons, colored light yellow, as seen in Figure 19 above. The first one is a 'Save Picture' button that will save the image currently enhanced and altered in its current state to the local hard drive for later use and perhaps further editing. The second one will enable annotation possibilities to the image if so requested and which can also be saved with button one. Detailed explanation on how to use the annotation is available further on.

## CADS Component Manual

ImageView [1]

Image: (Paste online URL, (Base64 text), or Load local Image File [Load File](#))



No image available

Border Color:



Size:

Style:

Image Title:

Image Caption:

Enable SciKit Image Processing [①](#):



Enable CSS Image Filter:



Extent:

[Cancel](#)[Submit](#)

Figure 20: The ImageView configuration form as it looks before any changes have yet been done.

When one opens the ImageView's configuration form the first time one is met with the view shown in Figure 20 above. The main basic fields available for the ImageView component, found in the ImageView configuration form are explained more in detail in the list below.

### ImageView Component's Form Fields

**Image:** This is where we apply an image to the component, either by loading from the local hard drive or pasting a URL (properly configured to allow 3<sup>rd</sup> party access and use) or even a Base64 string. The field will load and display the image if all is well with it, or if not display a message telling what is wrong. All web-based image formats are allowed, but SciKit image processing might work differently depending on the format and the number of channels contained within the image e.g. transparency.

**Border Color:** For display 'beautification' one can change the image border color.

**Border Size:** For display 'beautification' one can change the image border width size.

**Border Style:** For display 'beautification' one can change the image border style; e.g. solid or dashed.

## CADS Component Manual

**Image Title:** The title is auto generated from the image file name if such exist but may be manually altered at any time.

**Image Caption:** The image caption is auto generated to display the origin of the image file if available but may be manually altered at any time.

**Enable SciKit Image Processing:** This switch allows the user to enable SciKit Image processing to apply various algorithms (explained further on).

**Enable CSS Image Filter:** This switch allows the user to enable CSS filters to apply various filters (explained further on).

**Extent:** All components have width and height values encompassed in the extent group

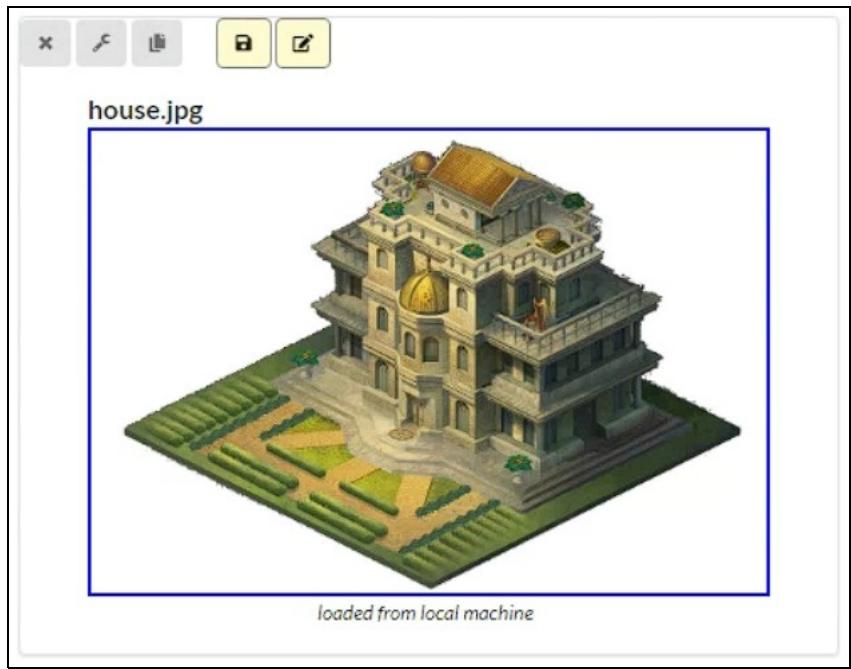


Figure 21: A simple image loaded from the local machine into the component.

When a simple image has been loaded and the simple basic configuration fields have been activated and submitted then the image component might look like something shown in Figure 21 above.

If one wishes to use the CSS filters and enables that in the ImageView configuration form an additional set of fields are opened up and displayed, as seen in Figure 22, that let the user control those filters. The CSS filters are mainly controlled by sliders that apply some form of numerical value, usually %. Since filters are done in the browser directly, there is a preview image available that show what the end result will look like if submitted and applied.

The CSS Filter fields available for the ImageView component, found in the ImageView configuration form when enabled are explained more in detail in the list below.

### *ImageView Component's Form Fields for CSS Filters*

---

**Gray Scale:** Change the color scale to be from 0% to 100% gray scale.

**Contrast:** Change the contrast to be from 0% (original) to 100% (full contrast change).

**Hue Rotate:** Change the rotation of the hue wheel from 0° (original hue) to 100° (back to original hue) with 180° being maximum hue change in the middle.

**Brightness:** Change the brightness from 100% (original) to either 0% (pitch black) or 500% (Very bright, almost full white).

**Invert:** Inverts the colors of the image from 0% (original) to 100% (fully inverted).

**Saturate:** Saturates the colors from 100% (original with no saturation) to either 0% (all saturation removed to gray) or 300% (maximum saturation of all colors).

## CADS Component Manual

**Sepia:** Change the sepia color level from 0% (original) to 100% (full sepia).

**Blur:** Makes the image blurry by mixing pixels close to each other from 0px (original) to 100px (extreme blurriness).

**Opacity:** Makes the more or less transparent, from an opacity of 100% (original, no transparency) to 0% (full transparent, no opacity).

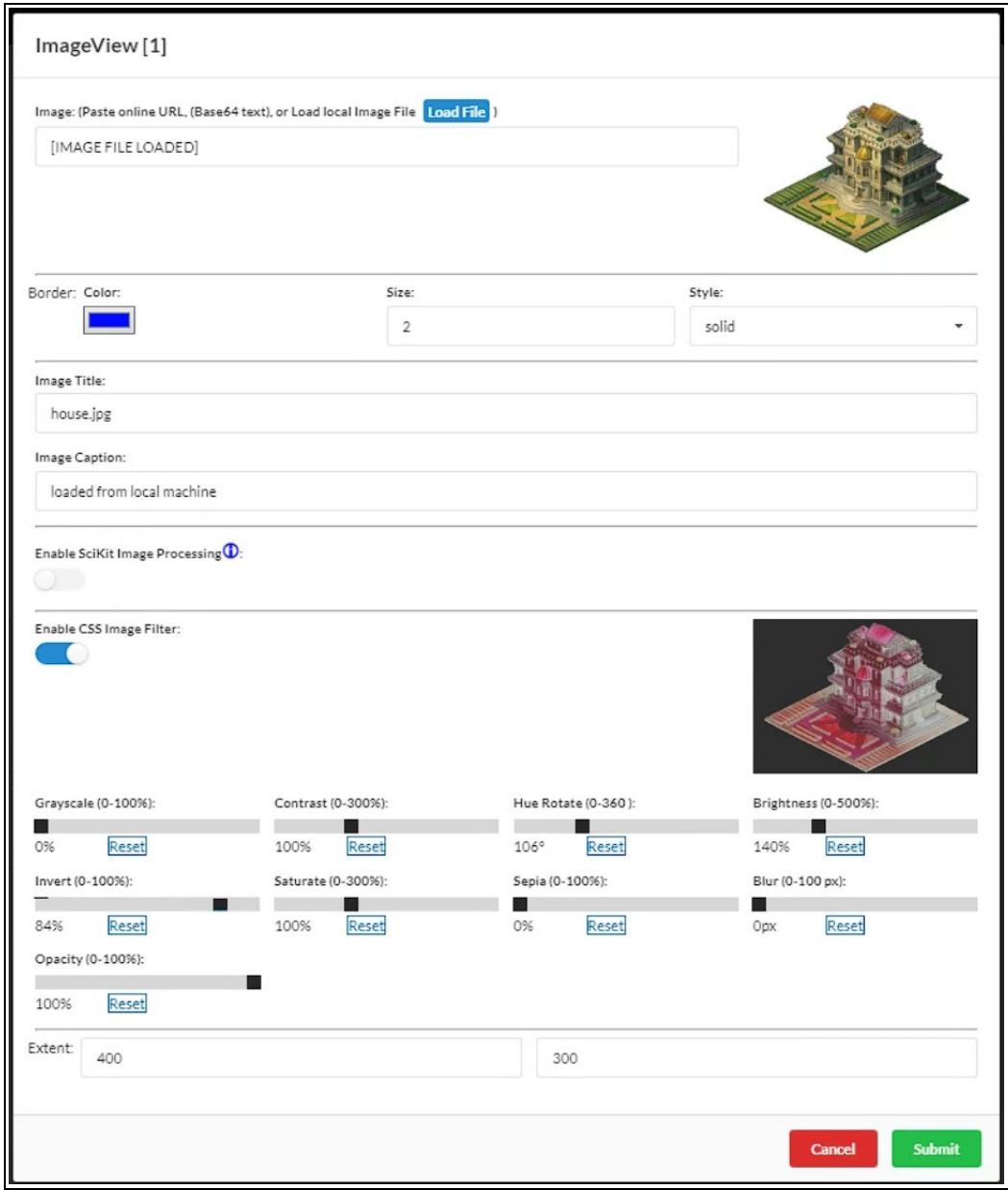


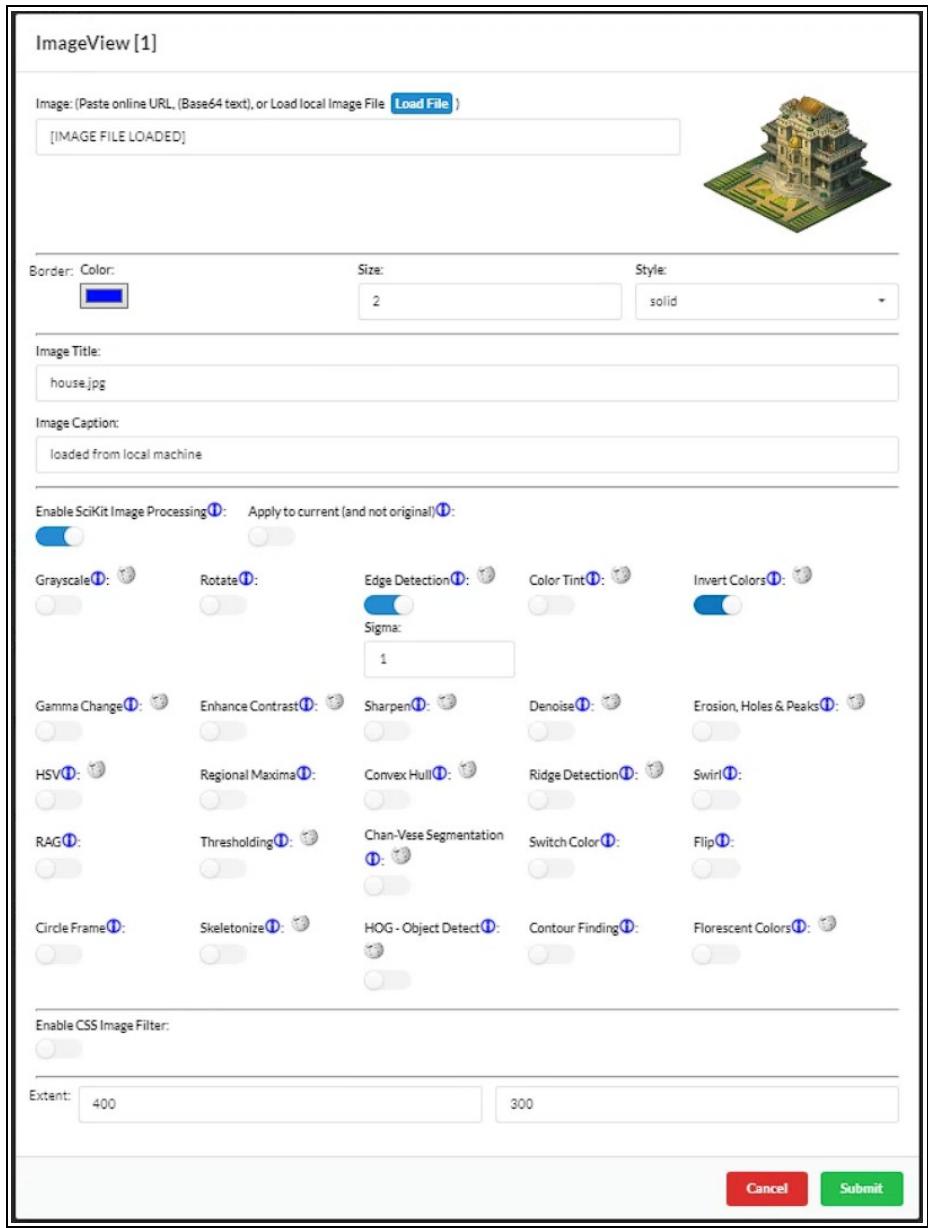
Figure 22: The ImageView component form focusing on CSS image filters that will alter the image.

Multiple filters may of course be applied at the same time all according to the end goal of the user.

When one enables the SciKit-image process, a long list of available algorithms becomes visible as seen in Figure 23. These algorithms are applied on the server side after form submit, so there is no preview image available. The top image will always show how the image currently look with all previously submitted changes included. But the ones currently being applied will not appear until after the server has finished manipulating the image in question.

The SciKit-Image processes are applied one by one In displayed list order as seen in the form, and that

## CADS Component Manual



*Figure 23: The ImageView Configuration form focusing on the SciKit-Image processing availability.*

may of course change the end result, since it does matter in what order any applied image manipulation happens. If the user is just using one algorithm or if the order of the list are perfectly fine then all is fine and no further control is needed. In Figure 24 one can see the result of applying the changes from Figure 23 on the image shown in Figure 21.

But if the user wants to apply multiple changes and they must be applied in a specific order, the approach would be to first apply the one or several that comes first, submit to the server and get the result. (perhaps save the interim result to the hard drive for safe keeping and even save the interim workspace for the same reason, though that is not necessary).

When the configuration form is opened up, our previous SciKit-Image settings have been stored and are now displayed in the form. But if we want to move on from our current state and apply additional changes without losing any recently applied ones, we turn on the switch to the right of SciKit-Image

## CADS Component Manual

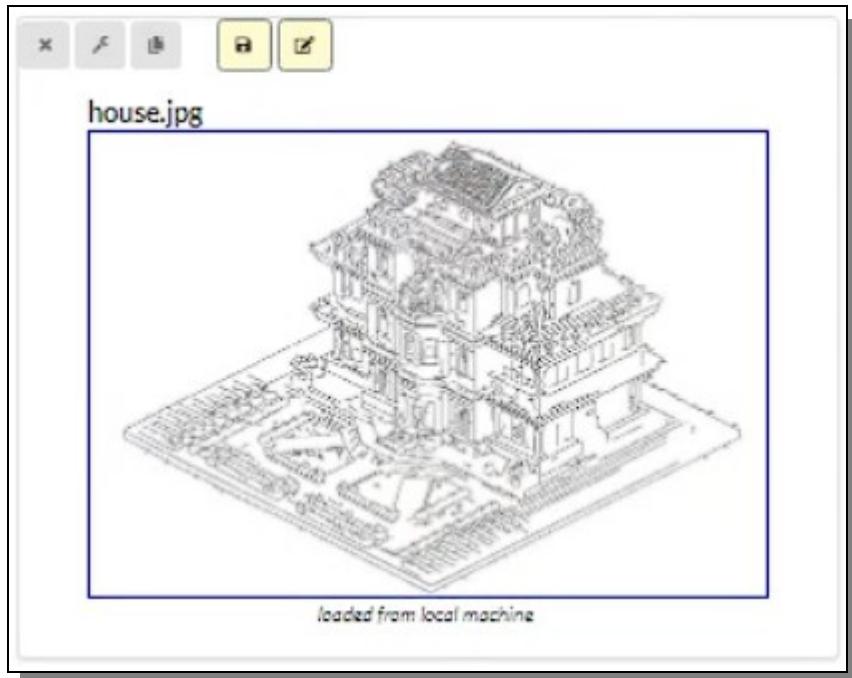


Figure 24: An example of some image manipulation done to an image using SciKit-Image.

enabling called "Apply to Current and not original". By doing that we can now reset all SciKit-image options and apply a new set, upon the ones we used before, even reapplying some of the algorithms a second or third time if that is what is wanted.

After submitting and server side have returned the result the user can continue this process any number of times.

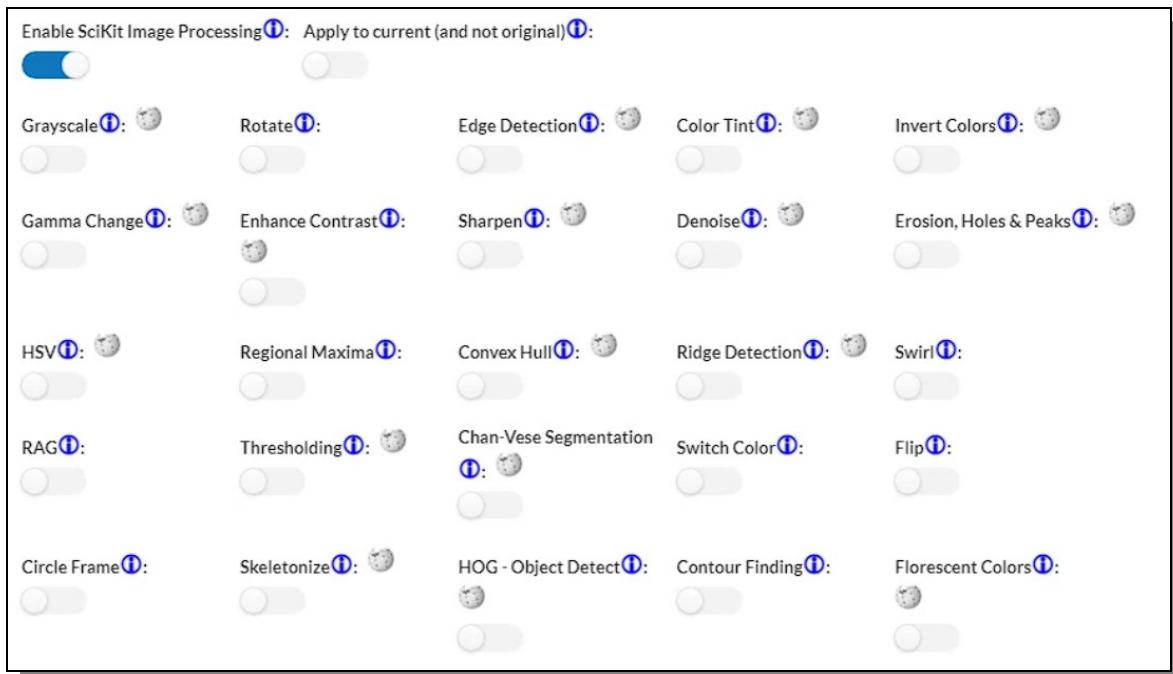


Figure 25: The available SciKit-Image processes that can be evoked to alter and enhance an image.

## CADS Component Manual

ddd

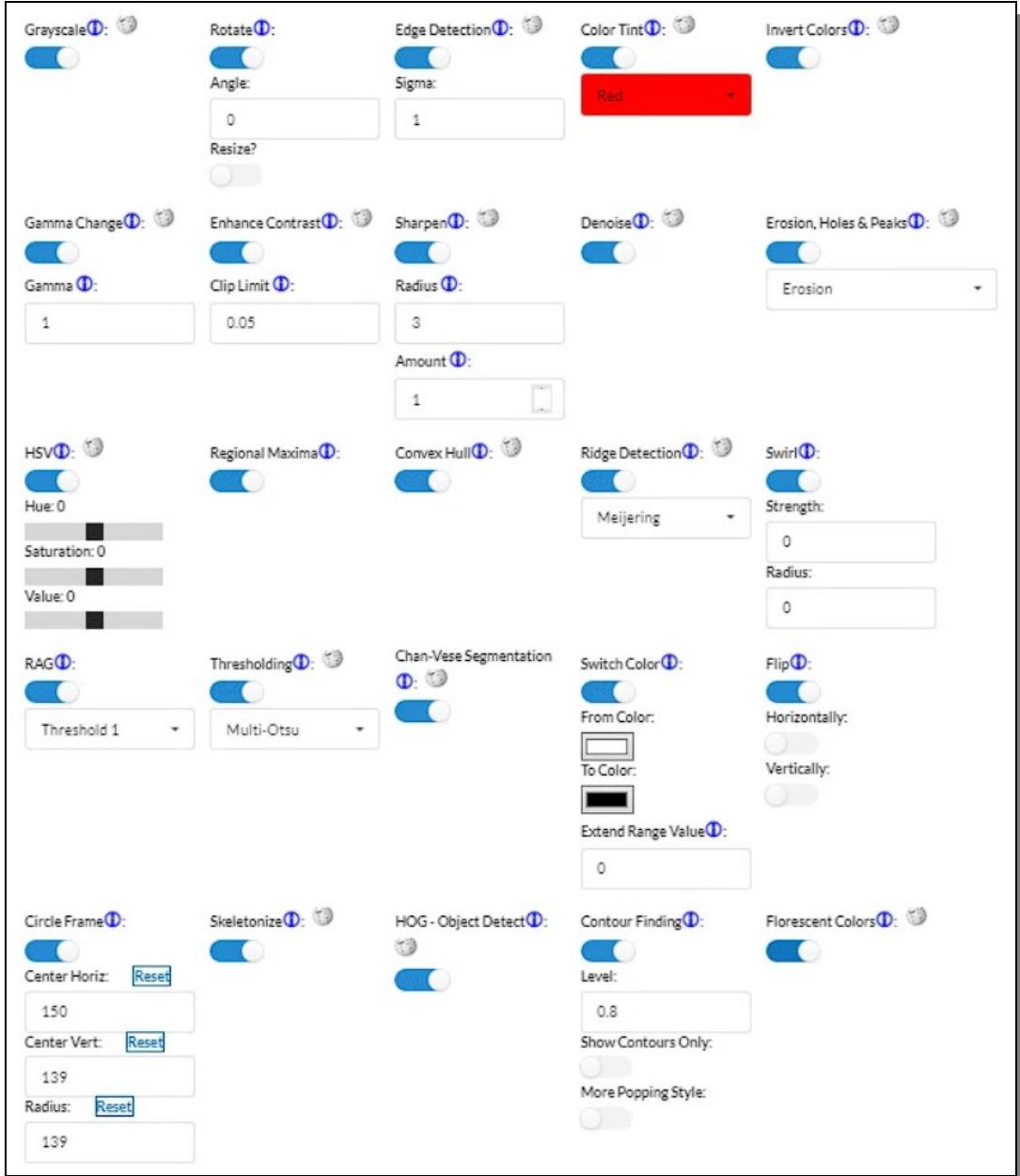


Figure 26: A detailed view on all available parameters for all available SciKit-Image processing one can apply.

In the configuration form there are several popup info boxes that better explains all available algorithms, including links to websites that in scientific details explains how it works if that is ever needed.

The list of available SciKit-image algorithms as seen in Figure 25 and with all its additional parameters when enabled as seen in Figure 26 above are explained in details in the list that follows.

### *ImageView Component's Form Fields for SciKit-Image Processing*

**Gray Scale:** Change the colors to be gray scale.

**Rotate:** Rotates the image  $x^\circ$ .

- **Angle:** The number of degrees to rotate

## CADS Component Manual

- **Resize?:** If the image should resize itself to fit its new size due to the rotation.
- Edge Detection:** Finds the edges of the image and remove all else. The edges will be drawn in white on black.
- **Sigma:** Gaussian Filter Standard deviation. 1 is default. Lower have more details and higher as less.
- Color Tint:** Applies a color tint to the whole image.
- **Color:** Which color to be used when tinting
- Invert Colors:** Inverts all the colors of the image.
- Gamma Change:** Makes an image darker or lighter by changing the gamma value.
- **Gamma:** 1 is unchanged, while lower is darker and higher is brighter.
- Enhance Contrast:** Make the contrast of the image more enhanced.
- **Clip Limit:** 0.05 is default and the higher value the stronger contrast is applied.
- Sharpen:** Makes the image much sharper and less blurry.
- **Radius:** 3 is default and it indicates the number of pixels next to target pixel that will be effected. More pixels, more sharper (and less softness)
  - **Amount:** 1 is default and the higher value the sharper it will become.
- Denoise:** Removes noise, discolored pixels and make it more smooth.
- Erosion, Holes & Peaks:** Flattens out darker areas (holes) and lighter areas (peaks) by 'eroding' both or just one of them in the Mode-setting .
- **Mode:** Erosion removes both holes and peaks, while peaks preserves holes and shadows and holes preserves peaks and highlights.
- HSV:** Hue, Saturation, Value color settings (instead of e.g. RGB-colors).
- **Hue:** The hue value of the HSV-colors
  - **Saturation:** The saturation value of the HSV-colors
  - **Value:** The value value of the HSV-colors
- Regional Maxima:** Preserves the pixel areas that have constant intensity and turns all the rest black.
- Convex Hull:** Attempts to create a white shape that would encapsulate the original foreground image and make the rest (background) black.
- Ridge detection:** Tries to find all ridges and edges of the image and remove all else.
- **Algorithm:** Meijering returns white ridges with black background and Hessian the opposite.
- Swirl:** Swirls the image in a spiral.
- **Strength:** How strong the swirl should be.
  - **Radius:** How big the radius of the swirls should be.
- RAG:** Regional Adjacency Graph create regions in the image and depending on the method divides them differently.
- **Method:** Threshold 1 or 2 or Merging gives slightly different results with different region borders.
- Thresholding:** Classic Image segmenting into simpler areas of light and dark.
- **Method:** Multi-Otsu tries to preserves colors while binary make the image pure black and white.
- Chan-Vese Segmentation:** Another type of image segmentation that works without parameters and is pure black and white.
- Switch Color:** Change one color of the image to another.
- **From Color:** This is the exact color (RGB) that should be changed
  - **To Color:** This is the exact color (RGB) that should be applied instead
  - **Extend Range Value:** This is the number of colors in direct relation with the From Color that should also be affected and changed
- Flip:** Flips the image either horizontally, vertically or both.

## CADS Component Manual

- **Horizontally?:** Enable if wanted.
- **Vertically?:** Enable if wanted.

**Circle Frame:** Creates a black circle frame on top of the image. Default is full circle, edge to edge, centered in the middle. But if one wish to change size and position it is available to do so.

- **Center Horiz:** The Horizontal center point of the circle frame
- **Center vert:** The Vertical center point of the circle frame
- **Radius:** The radius of the circle.

**Skeletonize:** If possible this algorithm will find the “skeleton” of the image and draw that and remove all the rest.

**HOG-Object detect:** HOG Object detection remove the image itself and replace it with short lines that make up the main object of the image as it was detected.

**Contour Finding:** If the image is big and complex, this might take some considerable time to finish, more minutes than seconds, and sometimes even time out.. It tries to find an image's main contours. Works best with well defined and simple images.

- **Level:** Works best with default value, but can be changed for different quality result
- **Show Contours only:** Remove the original image and just keep the contours.
- **More popping style:** Make the contour more visually strong.

**Florescent Colors:** Tries to replace all colors to be more fluorescent and popping for enhanced visibility.

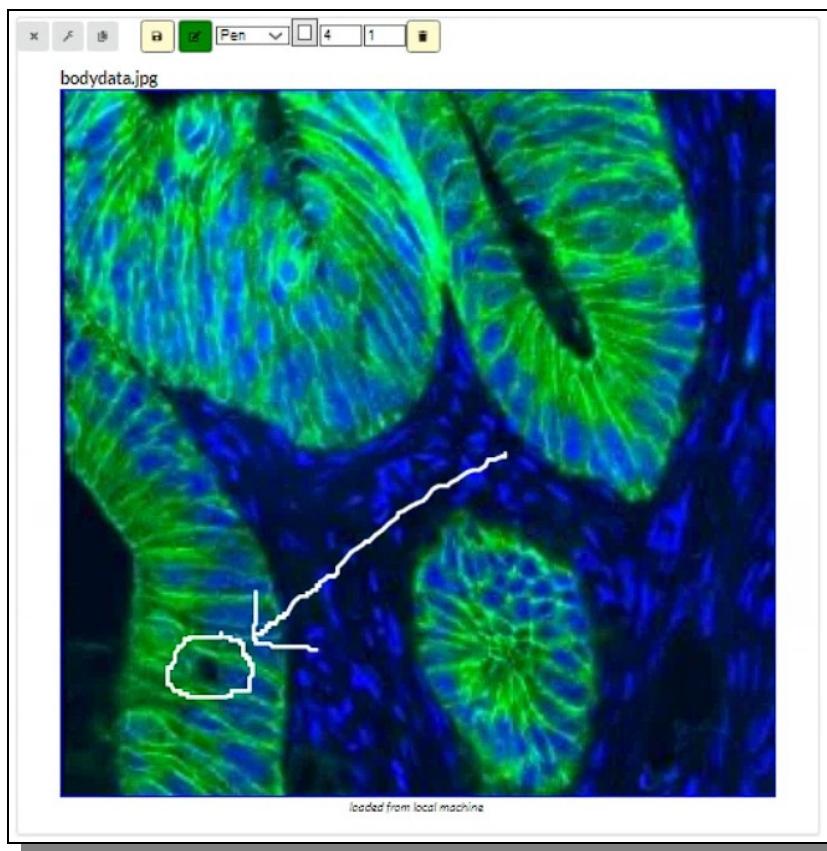


Figure 27: Here the annotation capabilities have been enabled and used for image enhancement.

When enabling Annotation of the component, the user can draw or erase lines with optional color, size and transparency as seen in Figure 27 above. Drawing is done by holding the left mouse button and moving it at the same time over the region of the image one wish to draw upon.

## Analysis Components

The Analysis components are those components that allows the user to enhance data in various complex ways that are more advanced than just *simple* visualization. These components may allow some form of data manipulation or have more advanced interfaces for re-focusing the view on specific parts of the data. Here we find such components like Histogram, Heat Maps and Parallel Coordinates.

### Parallel Coordinates

Parallel coordinates is a visualization method employed for representing individual data points across numerous performance metrics, with each metric aligning to a vertical axis and each data point being depicted as a connected series of points along these axes.

The parallel coordinates CADS component allows you to load all or some of the data columns as individual vertical axes. They can then be moved horizontally and put in any order the user want in order to detect interesting or revealing patterns in the data being studied.

In Figure 28 below you see an example of the parallel coordinates component with some data displayed.

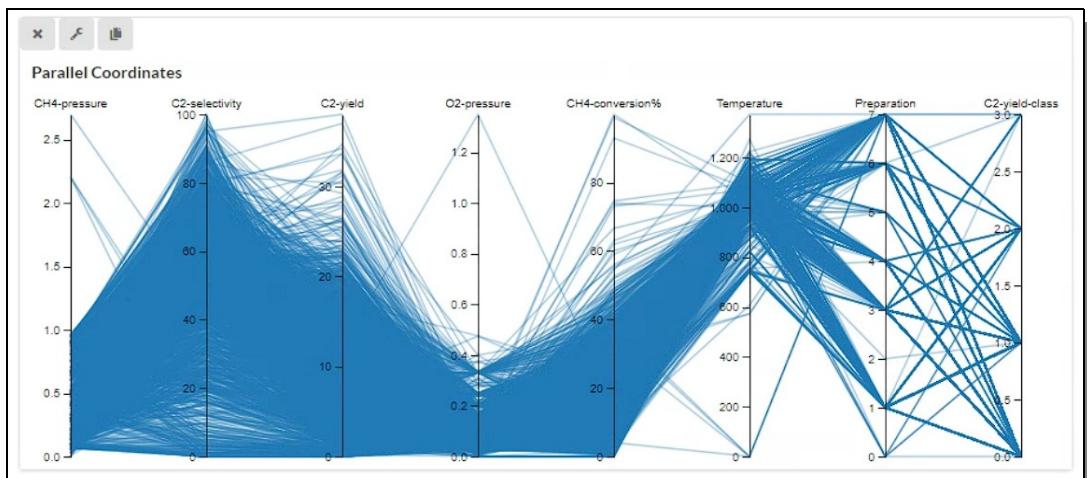


Figure 28: A CADS Parallel Coordinate component displaying the data for a set of columns.

To move one vertical axis, use the mouse to grab it by its name at the top and then drag it to where you want it and then let go and it will immediately jump into place. When the component is first loaded it will automatically contain all data columns but that can easily be changed in the configuration form.

Figure 29: The parallel coordinate configuration form is very slim and easy to use.

## CADS Component Manual

As seen in Figure 29 below the Parallel Coordinates configuration form contains an easy and small set of fields for controlling the view. These fields are explained more in detail in the list below.

### Parallel Coordinates Component's Form Fields

**Filter:** The Filter section in the form is for when you have assigned colors to the workspace as you can read about in the CADS User Manual, then you can use those colors to filter out specific data.

**Axes:** Select which data columns should have its own axis. (If none is selected all will be used).

**Extent:** All components have width and height values encompassed in the extent group

If one hover the mouse above any of the axis the mouse icon change to a cross-hair plus symbol. This indicates that the user can click and drag in order to select parts of the axis and thereby select parts of the data. When that is done all non selected data goes dim in order to highlight the data being focused on. This can be seen in Figure 30 below.

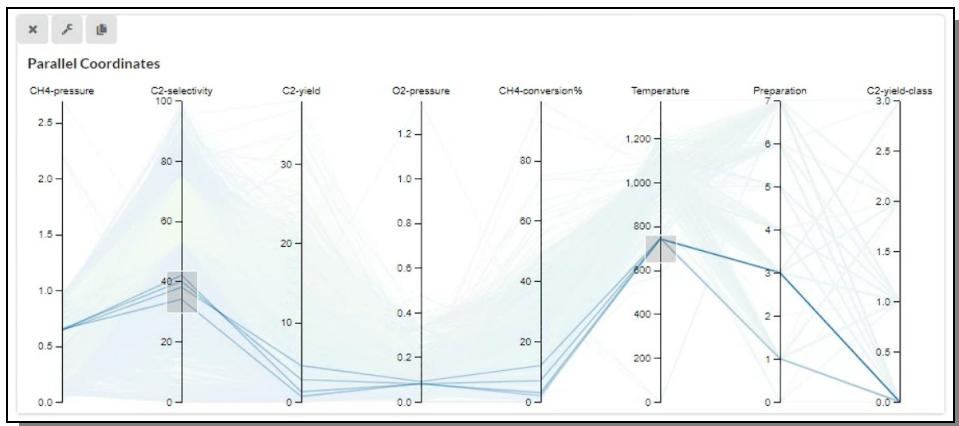


Figure 30: Some parts of some axis have been selected in order to focus on a particular part of the data.

In the above state the user can assign colors to the selected lines. In Figure 31 below we can see how the user have selected and assigned colors to the visualized dataset.

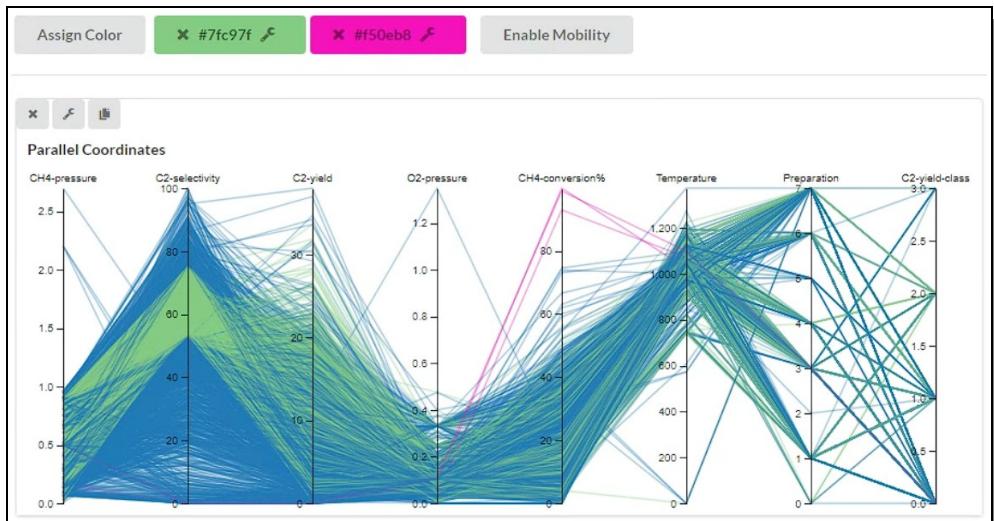


Figure 31: Here some of the data has been selected and colored as described above.

## Histogram

The CADS Histogram component does exactly what the name suggest, display the data as a histogram. One minor difference from other CADS components though, that one does not need to load a new component when adding another histogram for a new column, it is all included in the one component. So if one selects one data column to get a histogram for one get one histogram, but if one selects 4 different columns, 4 histograms are drawn. In Figure 32 below one can see four data columns have been selected to show four histograms of the data.

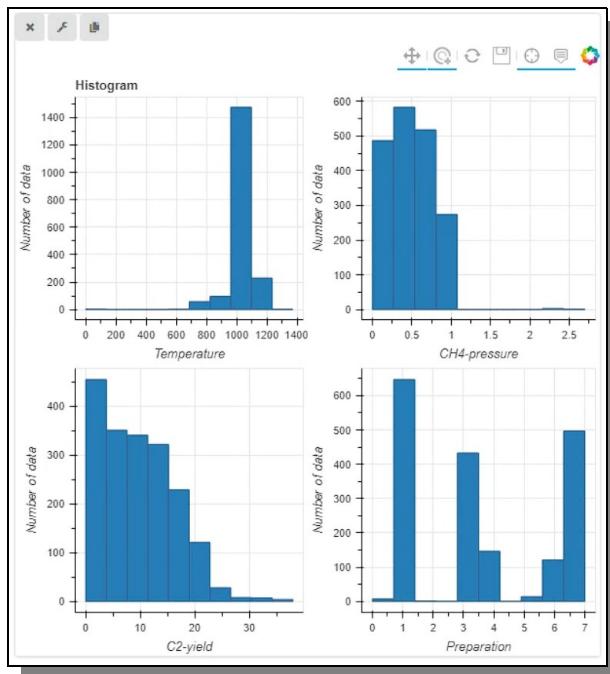


Figure 32: One histogram component displaying four histograms for four data columns.

As seen in Figure 33 below the Histogram configuration form contains an easy and small set of fields for controlling the view. These fields are explained more in detail in the list below.

### *Histogram Component's Form Fields*

**Target Columns:** This is where the user selects which data columns should get its own histogram.

**Number of bins:** The number of bins the histograms should be divided into.

**Extent:** All components have width and height values encompassed in the extent group

Figure 33: The Histogram configuration form could probably not be any simpler.

## Feature Importance (FD)

Feature Importance encompasses methods that compute a rating for all input features in a given model, with these ratings merely signifying the significance of each feature. A higher rating indicates that the particular feature will exert a more substantial influence on the model employed to forecast a specific variable.

In Figure 34 below we can see that each of the selected columns have been given a Featured Importance value and is displayed as a bar chart in our Featured Importance CADS component.

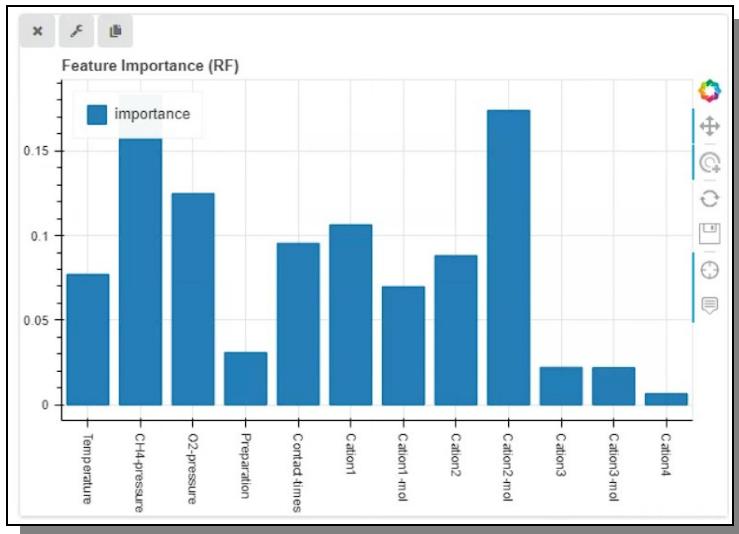


Figure 34: Displaying the featured importance for the selected columns in the data

As seen in Figure 35 below the Featured Importance configuration form contains an easy and small set of fields for controlling the view. These fields are explained more in detail in the list below.

### Feature Importance Component's Form Fields

**Featured Columns:** Here one select which data columns should be included in the feature importance calculation.

**Target Column:** This is the column towards the featured importance is calculated towards when finding which parameter makes the biggest changes.

Figure 35: The Featured Importance configuration form is easy and straight forward.

**Extent:** All components have width and height values encompassed in the extent group

The featured importance calculations are made on the server and might take some time if the data is very big.

## CADS Component Manual

### Heat map

The CADS Heat map component basically works and look like one would expect a heat map to work and look. In Figure 36 below you see an example of how some data (Iris) could look when displayed in the heat map. Of course the visualization quality is of course based on what kind of data is being used for the component in question.

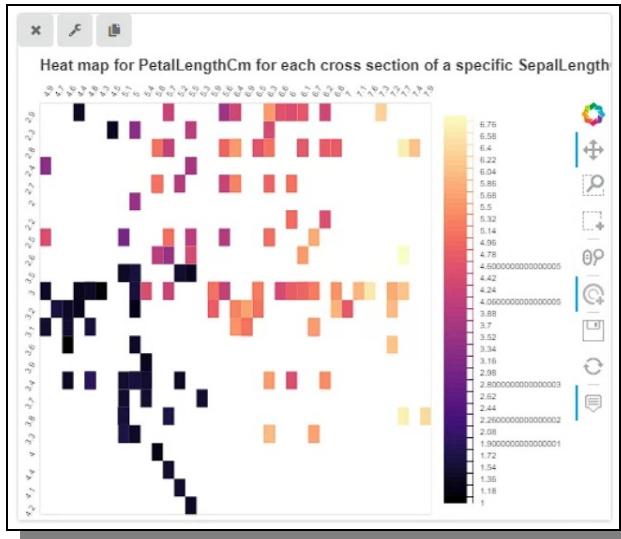


Figure 36: A CADS Heatmap component displaying data the heatmap way.

The fields available for the heat map component and which are found in the Heat map configuration form are explained more in detail in the list below.

#### Heat Map Component's Form Fields

**Horizontal Data:** Which data column that should be used for the horizontal X-axis

**Vertical Data:** Which data column that should be used for the vertical Y-axis

**Color Data:** Which numerical data column that should be used applying a color to the heat map cell.

**Extent:** All components have width and height values encompassed in the extent group

**Color Palette:** Which palette of colors that should be used when coloring the heat map cells.

In Figure 37 below you can see how the heat map configuration form look and where each field is edited.

Figure 37: The Heat Map Configuration form should not be any problem to edit for a good result.

## Pairwise Correlation

Pairwise correlation is a statistical method used to measure the relationship between two variables in a dataset. It calculates a correlation coefficient to determine the strength and direction of the relationship. It helps identify associations between variables, aids in variable selection for predictive modeling, supports data exploration, and can be used for hypothesis testing. However, it's essential to remember that correlation does not imply causation, as causation requires additional evidence and experimental design.

In Figure 38 below we see the CADS Pairwise correlation component visualizing a set of data variables in a classic way.

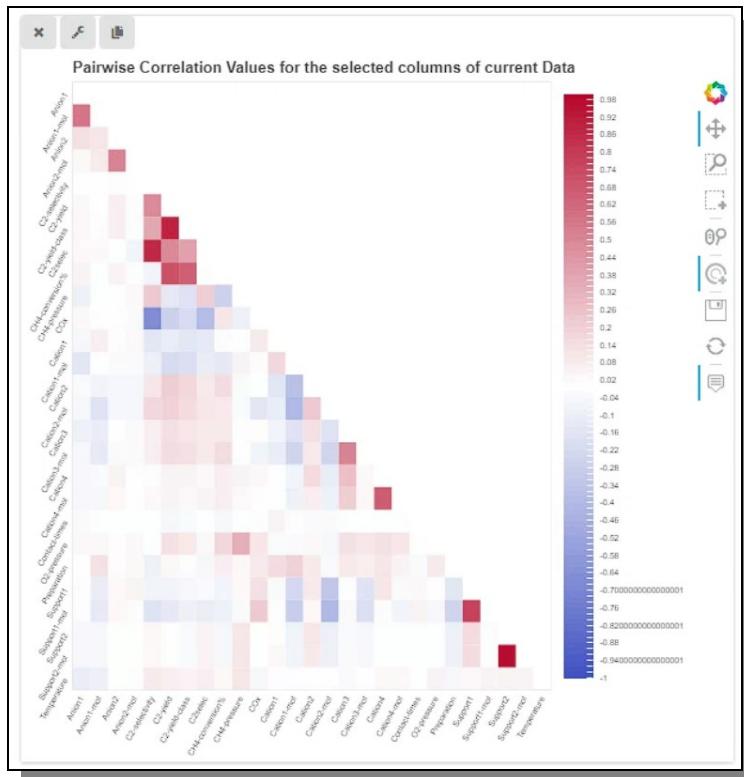


Figure 38: A Pairwise Correlation visualization is also available in CADS.

The fields available for the pairwise correlation component and which are found in the pairwise correlation configuration form are explained more in detail in the list below.

### Pairwise Correlation Component's Form Fields

**Featured Columns:** Which data columns one wish to include in the pairwise correlation visualization.

**Select All Columns:** In case one one to select all data variables (columns) this switch make it fast and easy instead of picking them one by one.

**Enable Half-mask:** This is on by default since that is the classic look for pairwise correlation, but if one want to view the data in a more heat map way one may just turn it off.

**Select Color Range:** The colors used are a default classic palette usually used for this kind of visualization, but if one want a brighter, heat map-stylish and more contrast set of colors then this switch will enable that.

**Extent:** All components have width and height values encompassed in the extent group

## CADS Component Manual

In Figure 39 below you can see how the pairwise correlation configuration form look and where each field is edited.

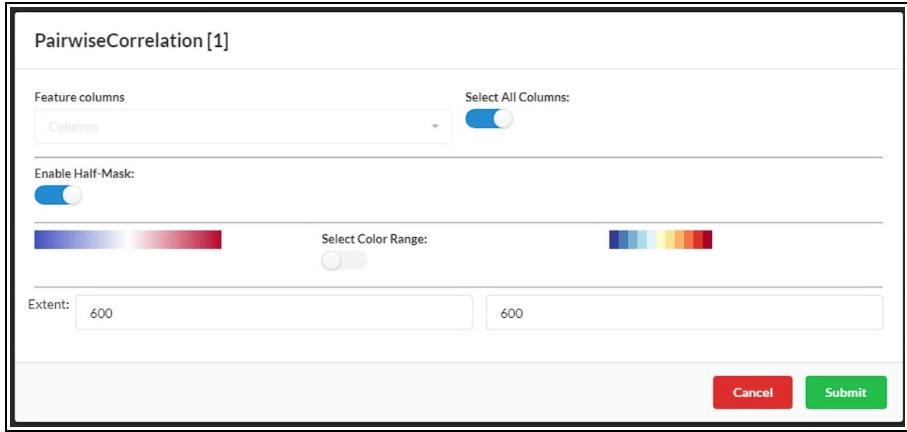


Figure 39: The pairwise correlation configuration form is as expected simple and easy.

## NodeGraph

A node graph, also known as a network graph, is a mathematical and visual representation of a collection of interconnected objects. In a node graph nodes are individual entities or points in the graph. They represent objects, elements, or entities of interest, such as people, places, or items. Edges are the connections or relationships between nodes. They depict how nodes are linked to one another. Edges can be directed (with an arrow indicating a one-way connection) or undirected (indicating a two-way connection), but the CADS NodeGraph component is only dealing with undirected relationships.

Node graphs are used in various fields, including computer science, social network analysis, biology, transportation planning, chemistry, and more. They are particularly useful for modeling and visualizing complex relationships and systems.

When one loads a CADS Node Graph component it will not yet be connected to any data, but still show a simple sample node network graph for learning purpose, as can be seen in Figure 40 below.

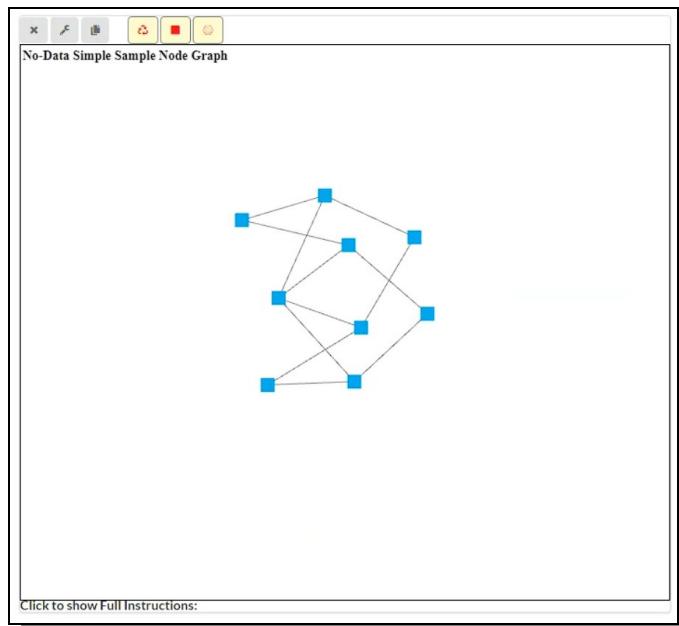


Figure 40: A simple sample network node graph in CADS

## CADS Component Manual

This CADS component have three custom, yellow colored, buttons alongside the normal component buttons.

The first one is for pinning the nodes. When a network graph is being drawn, it is at all time calculating its position in reference to other nodes and the relationships it has and how strong that relationship is. This will make all nodes moving until it stabilize in a perfect spot. For small graphs, that might take less than a second, but for larger graphs it could seemingly go on forever. To stop, or more exactly pause, that from happening the user can click the first 'Toggle Node Pinning' and all nodes will then freeze in their current position and be still. If one wish to allow the network to continue the self calibration of its node positions again, one just press the same button once again to toggle it off.

When the nodes are still they can be manually moved by the user with a simple mouse drag and drop operation.

Good to know is at the bottom of the component there is a set of instructions for how to interact with the graph so that one does not need to go back to this manual all the time (Figure 41).

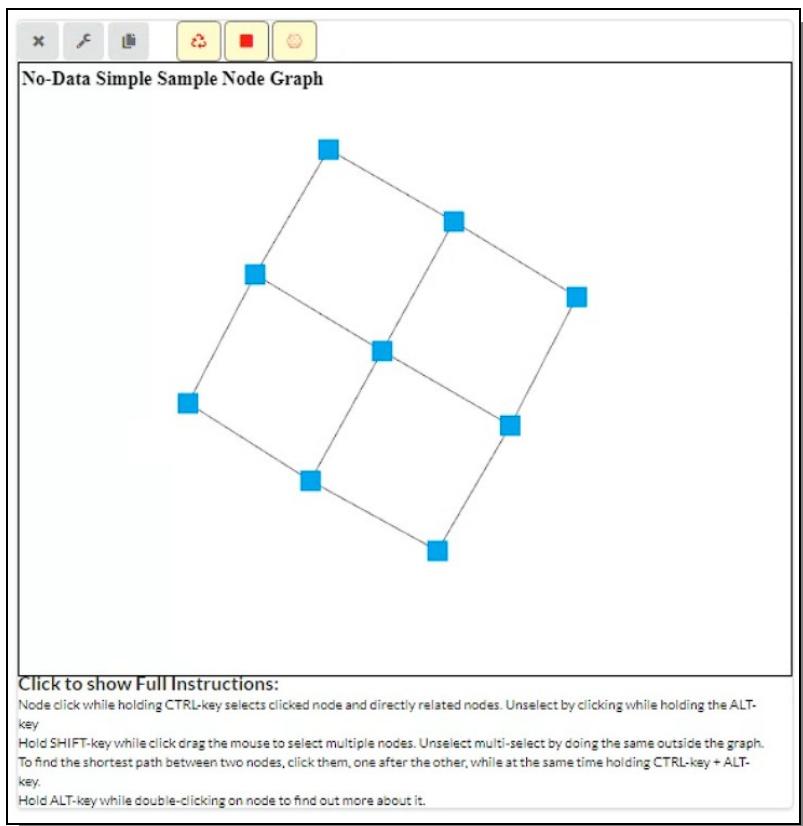


Figure 41: The Nodes evenly placed after they have been moved by the user.

A node graph network is a very specific type of data, so applying any sort of data to it, will not always be useful and work as intended, but if one has typical node data, with source nodes, target nodes and perhaps if available link weights, than this is the component to use.

The fields available for the NodeGraph component and which are found in the NodeGraph configuration form are quite extensive, but are explained more in detail in the list below.

### ***Node Graph Component's Form Fields***

**Source Node Column:** The Nodes that are found in the source node column of the data.

**Target Node Column:** The Nodes that are found in the target node column of the data.

## CADS Component Manual

**Link/Edge Weight Column:** The weight values for the connecting link/edges that are found in the data.

**Select All Columns:** In case one needs to select all data variables (columns) this switch makes it fast and easy instead of picking them one by one.

**Link Settings – Base Length:** The default base length in pixels for a link between two nodes, before it has been manipulated for weights and other network graph settings.

**Link Settings – Min Length:** The smallest length a link between two nodes can ever have after it has been manipulated for weights and other network graph settings.

**Link Settings – Static Color:** The color the links are drawn with (as long as they are not curves which is always blue).

**Link Settings – Opacity:** The opacity of the link lines.

**Link Settings – Bezier Curve Enabled:** If enabled the links will be drawn as curved arcs instead of straight lines (These curves will also only use blue color and ignore the static color setting).

**Layout – SpringCoeff:** Hook's law's coefficient that will adjust how springy the link between two nodes. A value of 1 makes it solid and lower values make it more bouncy and stretchy.

**Layout – DragCoeff:** Drag force coefficient. Used to slow down the system, thus should be less than 1. The closer it is to 0 the less tight the system will be, and more spread out

**Layout – Gravity:** Coulomb's law coefficient. It's used to repel nodes thus should be negative. If you make it positive, nodes start to attract each other like true gravity.

**Layout – Theta:** Theta coefficient from Barnes Hut simulation. Ranged between 0-1. The closer it's to 1 the more nodes algorithm will have to go through. Setting it to one makes Barnes Hut simulation no different from brute-force forces calculation (each node is considered)

**Layout – TimeStep:** Default time step (dt) for forces integration.

**Node Settings – Static Color:** The static color of the nodes.

**Node Settings – Static Size:** The static size color of the nodes.

**Node Settings – Enable Color Gradient:** When enabled, the colors of the nodes will depend on the amount of links and the weights of those links, ignoring the static color setting.

**Node Settings – Enable Size Gradient:** When enabled, the size of the nodes will depend on the amount of links and the weights of those links, ignoring the static size setting.

**Node Settings – Opacity:** The opacity of the nodes.

**Node Settings – Shape Type:** The shape of the node, that can be either square, round or a static image.

**Node Settings – Round Shape Border Enabled:** When enabled and the shape is set to round, this will give the node a black border.

**Node Settings – Image Shape Image URL:** When the shape is set to image then this will be the url to the image being used.

**Background Color:** The background color of the graph component (white by default).

**Text Color:** The text color used in the graph component (black by default).

**Extent:** All components have width and height values encompassed in the extent group

In Figure 42 below you can see how the Node Graph configuration form look and where each field is edited.

How to properly set the various parameters in the configuration form, is a true art form and science all by itself, so that is not something we can tell you, but by testing various settings you might find the most optimal one for your data.

## CADS Component Manual

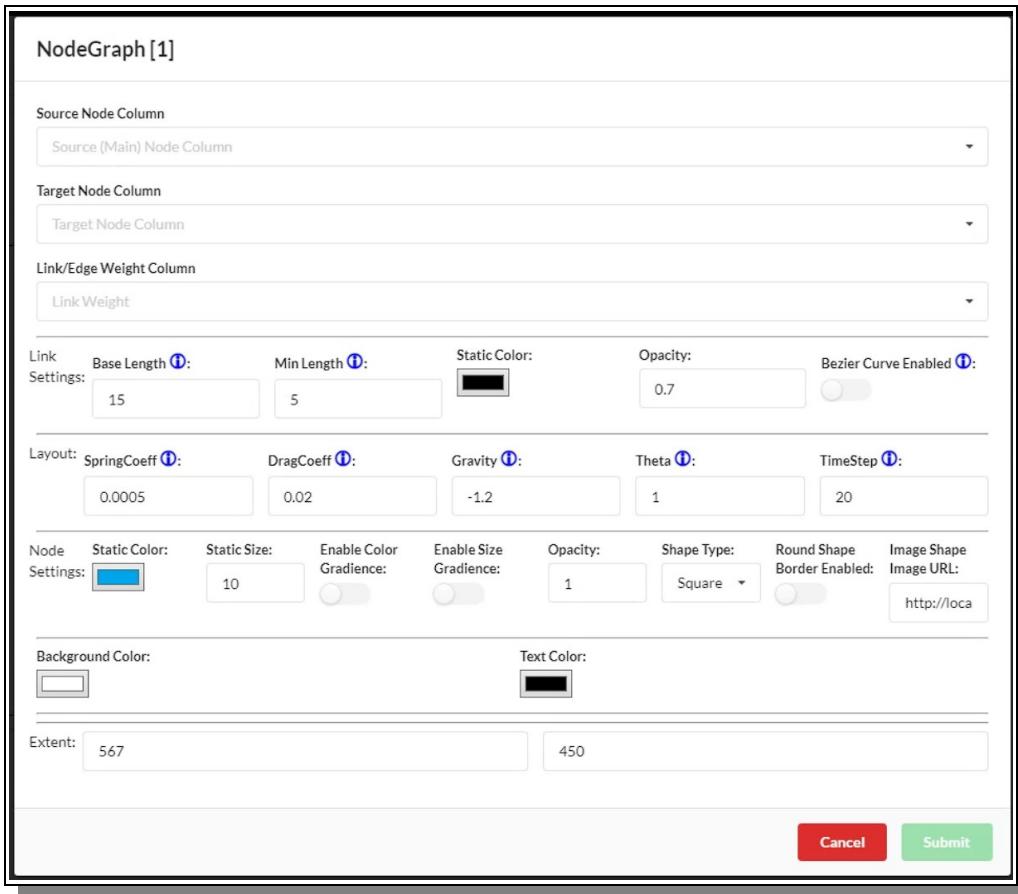


Figure 42: The Node Graph Configuration Form and all its available settings to control the network.

If the network data is prepared correct and the settings in the configuration form is left as is a graph will be shown in the component that can be manipulated and interacted further with. In Figure 43 an example of such a network graph can be seen.

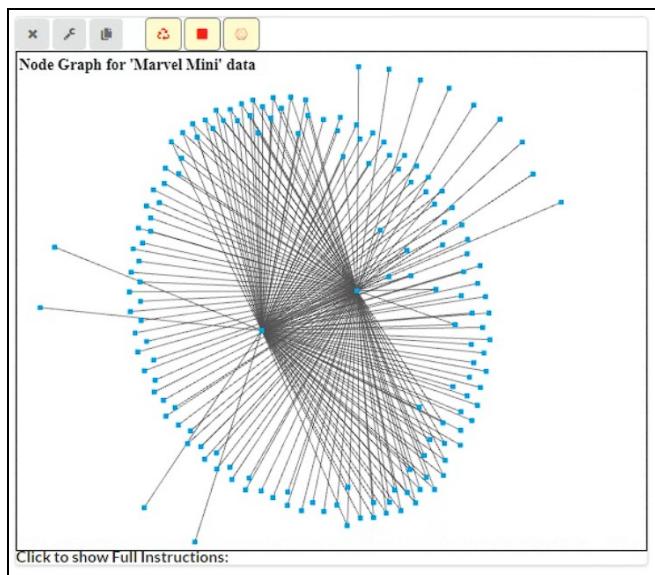


Figure 43: A proper node graph drawn based on real network data

## CADS Component Manual

In the Figure 44 below, we see another network graph loaded with a range of other settings made inside the configuration form.

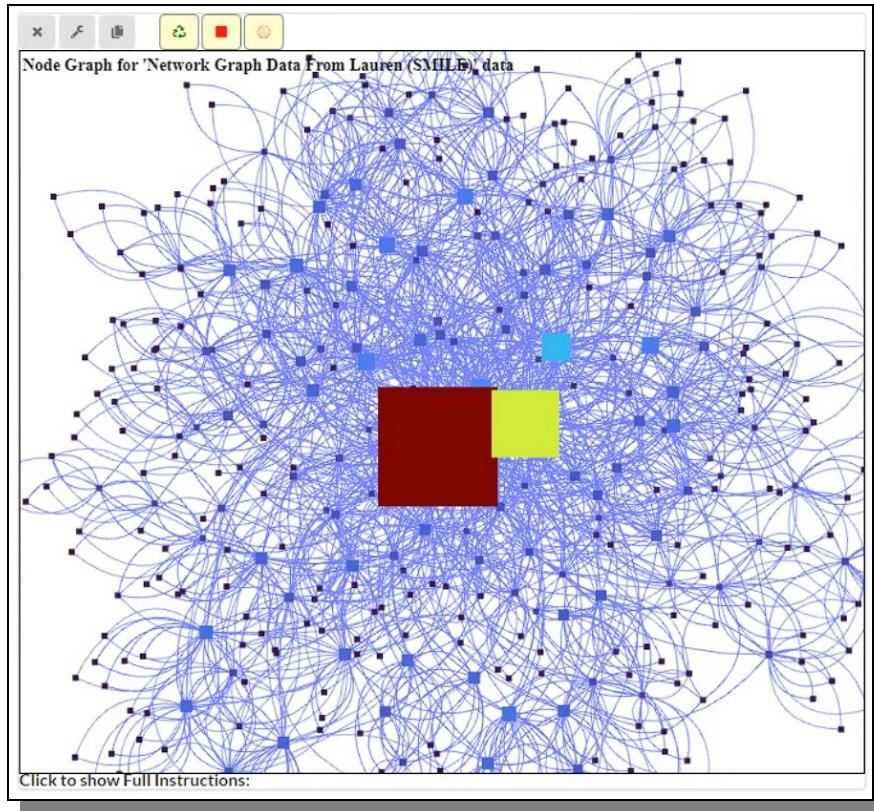


Figure 44: A network graph using a bunch of the available settings in the configuration form, such as e.g. bezier curve and color and size gradiance.

The second custom yellow button for this component, toggles whether the name of the node should be displayed or not as can be seen in Figure 45 below.

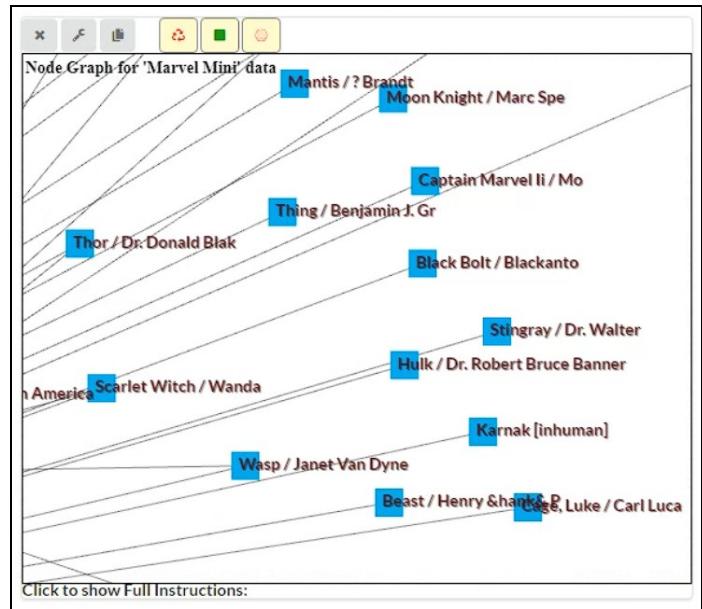


Figure 45: Displaying the names for each node in the network graph.

## CADS Component Manual

The third and final custom yellow button toggles the visibility of the names for each link (which is basically the combined names of both connected nodes) and can be seen in Figure 46 below.

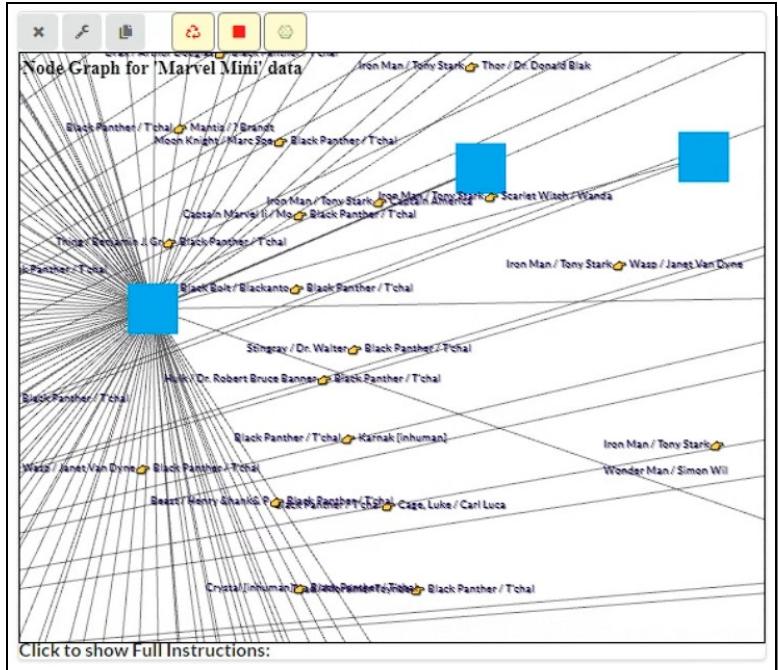


Figure 46: Displaying the names for every link between nodes in the network graph.

In Figure 47 below you can see how a very large and complex network graph have been loaded and displayed. Since this component is using OpenGL and VivaGraph to display its nodes and edges, it does so very fast and well even with large data.

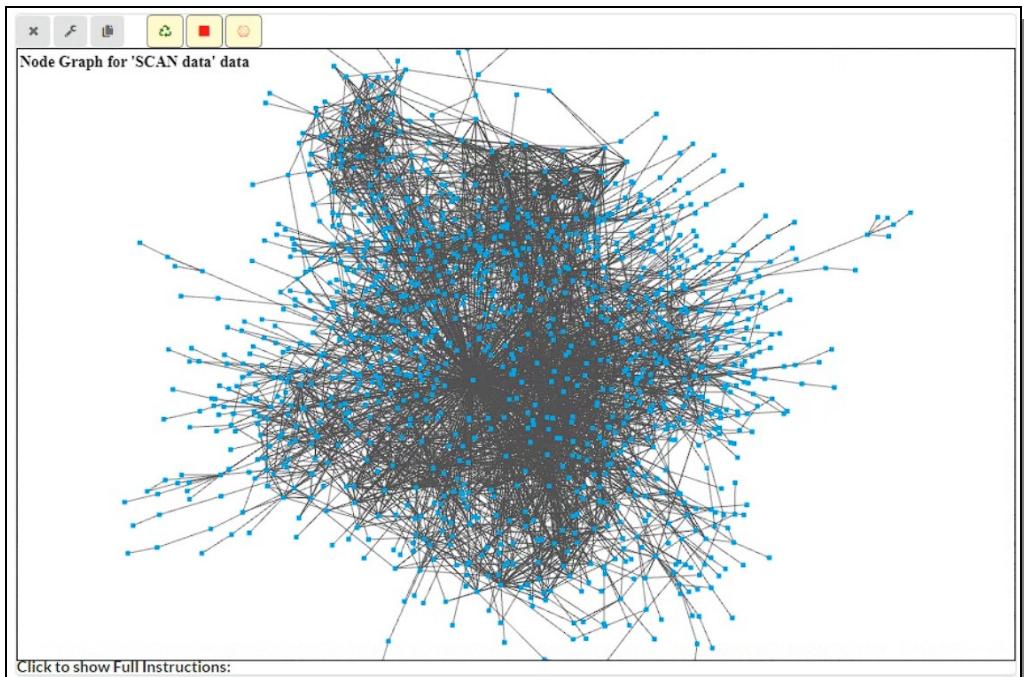


Figure 47: A really large node network graph loaded into the component with thousands of nodes.

## CADS Component Manual

The user can interact with the network in various ways, as also explained in the instructions below of the component as mentioned earlier.

One way is to select a node and all its closest neighbors by clicking the node while holding the CTRL key. The main clicked node will become yellow and all neighbors will be orange. This can be seen in Figure 48 below

To deselect, one click outside the graph while holding the ALT key.

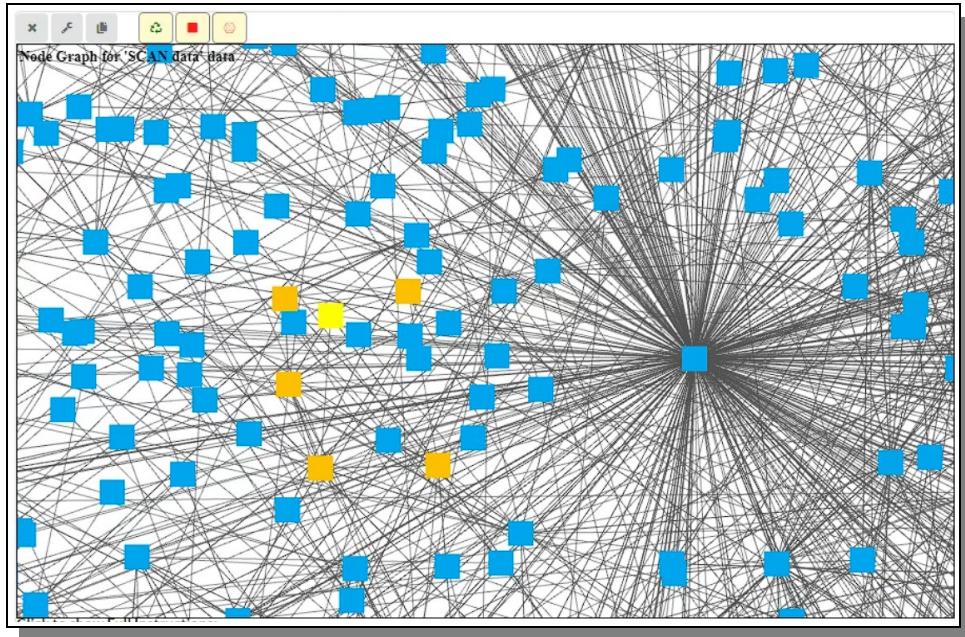


Figure 48: A node and its neighbors have been selected.

Holding the SHIFT key while click drag the mouse, one can select multiple nodes at the same time as can be seen in Figure 49 below. To deselect such a selection just redo the operation on the side of the graph and selecting nothing.

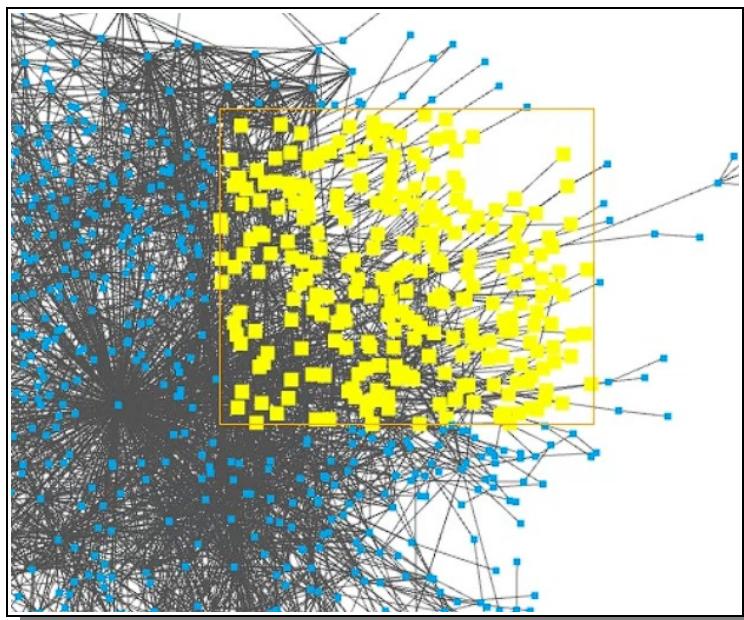


Figure 49: Multiple select of many nodes is also possible inside the component.

## CADS Component Manual

Another feature available in the interaction section of the live graph is to calculate and display the shortest path between two selected nodes. To do that one holds the CTRL and ALT key at the same time and then mouse click first the start node followed by a click on the end node. As soon as the final click is made the shortest path will be calculated and displayed as can be seen in Figure 50 below.

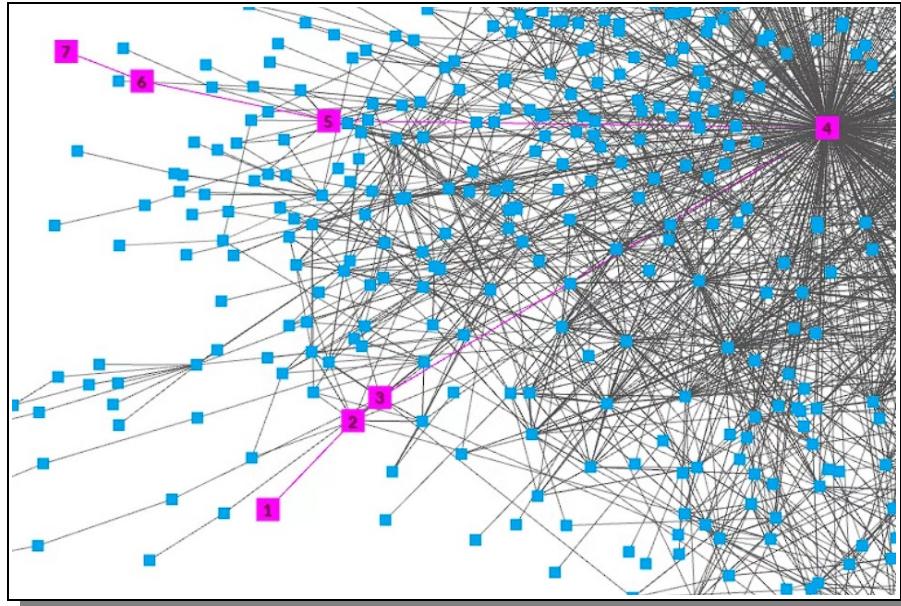


Figure 50: Calculating and showing the shortest path between two nodes in the graph.

If one double click a node while holding the ALT key, the name of the node will be used to do a Google search and open up in a separate tab in the browser to find out more about the node content if possible.

These interactions are not solid and will therefore not be saved when saving the workspace. Only changes made in the configuration form are saved inside the workspace.

## Machine Learning Components

The Machine Learning components are those components that have various ML features and other AI related capabilities. When using these components you train models based on the current used data which can often also be saved for the Prediction Section of the CADS website. Here we find such components such as Regression and Classification but more are sure to come in the future.

### Regression

Regression in machine learning is a supervised learning technique used for modeling the relationship between a dependent variable (target) and one or more independent variables (predictors or features). The goal of regression is to predict a continuous numeric output, which can represent various types of data, such as prices, scores, quantities, or any value within a continuous range.

In a regression problem, you typically have a dataset with historical data that includes both input features and corresponding target values. The algorithm learns from this data to create a mathematical model or function that maps the input features to the target variable. Once trained, the regression model can be used to make predictions on new, unseen data.

There are a wide range of regression algorithms available and the choice of regression algorithm depends on the nature of the data and the specific problem you are trying to solve.

The CADS Regression component is a small powerhouse of easy to use options for training a model on a set of data.

## CADS Component Manual

When first loaded the view is empty, as seen in Figure 51 below, since before showing anything the user need to configure how the regression should be done, so the first thing to do is to enter the configuration form and set the available parameters.

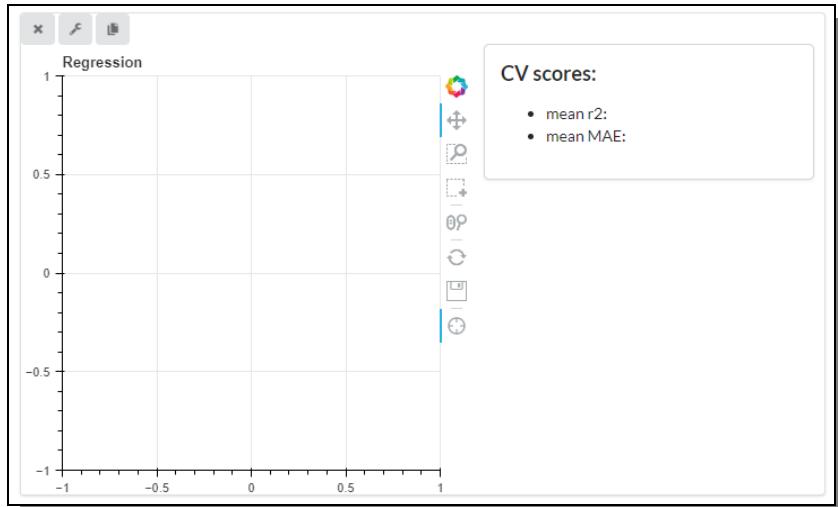


Figure 51: A newly loaded, empty Regression component ready to be used.

The fields available for the Regression component and which are found in the Regression configuration form are quite extensive and complex in some way, but they are grouped and organized in such a way that it should be fairly easy to know what to do even for a layman and they are explained more in detail in the lists below.

### ***Regression Component's Form Fields (global in all settings)***

**Method:** Which Regression algorithm/method to use.

**Featured Columns:** Which feature columns in the data that should be used when training the model.

**Target Column:** Which target column, the value we are seeking to predict, that should be used/targeted when training the model.

**CV Method:** Which cross validation method to use when training the model.  
(TrainTestSplit or KFold)

**Extent:** All components have width and height values encompassed in the extent group

### ***Regression Component's Form Fields (local based on selected methods)***

**SVR Parameters - C:** A regularization parameter used to set the tolerance of the model to allow the misclassification of data points in order to achieve lower generalization error. Higher the value of C, lesser is the tolerance. Smaller the value of C, larger is the tolerance of misclassification.

**SVR Parameters - gamma:** The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The lower values of gamma result in models with lower accuracy and the same as the higher values of gamma. It is the intermediate values of gamma which gives a model with good decision boundaries.

**RandomForest Parameters - random\_state:** Controls the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node.

**RandomForest Parameters - n\_estimators:** Represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably, therefore the sweet spot is needed.

## CADS Component Manual

**ExtraTrees Parameters - random\_state:** Like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement.

**ExtraTrees Parameters - n\_estimators:** Like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement.

**MLP Parameters - random\_state:** Controls the randomness.

**MLP Parameters - max\_iter:** Maximum number of iterations.

**KernelRidge Parameters - alpha:** Regularization strength; must be a positive float. Regularization improves the conditioning of the problem and reduces the variance of the estimates. Larger values specify stronger regularization.

**CV Method – TrainTestSplit – test\_size:** If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples.

**CV Method – KFold – n\_splits:** Number of folds. Must be at least 2.

In Figure 52 below you can see how the Regression configuration form look and where each field is edited.

How to properly set the various parameters in the configuration form, is a true art form and science all by itself, so that is not something we can tell you, but by testing various settings you might find the most optimal one for your data.

The screenshot shows the 'Regression [1]' configuration form. It includes fields for 'Method' (set to 'RandomForest'), 'Feature columns' (listing 'Preparation', 'CH4-pressure', 'Temperature', and 'O2-pressure'), 'Target column' (set to 'C2-yield'), and 'RandomForest Parameters' (with 'random\_state' at 0 and 'n\_estimators' at 100). Under 'Cross Validation', the 'CV Method' is set to 'TrainTestSplit' with 'test\_size' at 0.2. At the bottom, there are 'Extent' fields for 'Width' and 'Height', a 'Save model' button, and 'Cancel' and 'Submit' buttons.

Figure 52: The Regression CADS component configuration form being edited

When the configuration form is submitted it will contact the server in order to do the calculations requested using the Scikit Learn library. Depending on the data size and the configuration setting this might take some time and a waiting indicator will show up to visualize that, though the exact time will of course not be known in advance. Whether it takes 20 seconds or 20 minutes is a matter of ML

## CADS Component Manual

experience, but the web-server will time out after a certain time, so keep the data size and settings such so that the time is not becoming extreme.

In Figure 53 below you can see the result of a Random Forest Regression on some chemical data plotted in a familiar way, including Cross validation scores.

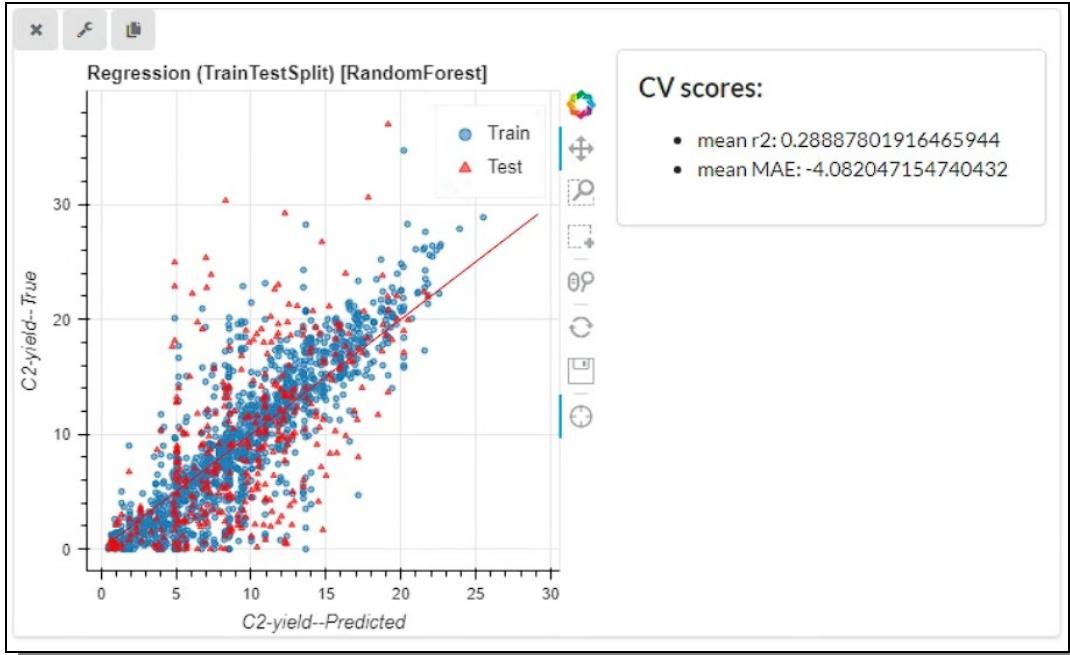


Figure 53: A Random Forest Regression visualization result in the CADS component

At the bottom of the configuration form, as can be seen in Figure 52 in the previous page, there is a 'Save Model' button, which is disabled at start, but after the training is done and finished becomes enabled. If one press that button a popup window shows which asks the user to give this model prediction a name, and after doing so and clicked 'Save', the server will once again be contacted and the loading indicator show up. When it is done, the model will then be available for use under the Prediction tab in the top main menu, as detailed explained in the CADS User Manual.

## Classification

Classification in machine learning is the process of categorizing data into predefined groups or classes based on its characteristics or features.

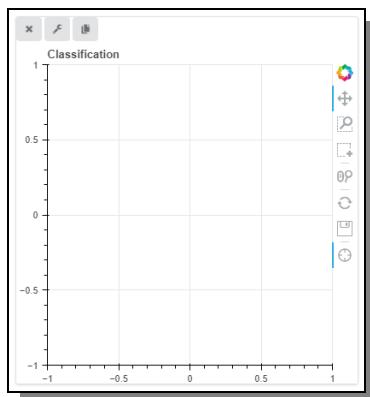


Figure 54: A newly loaded, empty Classification component ready to be used.

## CADS Component Manual

The CADS Classification component is like the Regression component a small powerhouse of easy to use options for training a model on a set of data.

When first loaded the view is empty, as seen in Figure 54 above, since before showing anything the user need to configure how the regression should be done, so the first thing to do is to enter the configuration form and set the available parameters.

The fields available for the Classification component and which are found in the Classification configuration form are quite extensive and complex in some way, but they are grouped and organized in such a way that it should be fairly easy to know what to do even for a layman and they are explained more in detail in the lists below.

### *Classification Component's Form Fields (global in all settings)*

---

**Filter:** If color tags are created as explained before and in the main User Manual in combination with data selection in the chart, then by using filters one can hide all other data from view.

**Method:** Which Classification algorithm/method to use.

**Featured Columns:** Which feature columns in the data that should be used when training the model.

**Target Column:** Which target column, the value we are seeking to predict, that should be used/targeted when training the model.

**Extent:** All components have width and height values encompassed in the extent group

### *Classification Component's Form Fields (local based on selected method)*

---

**RandomForest Parameters - random\_state:** Controls the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node.

**RandomForest Parameters - n\_estimators:** Represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably, therefore the sweet spot is needed.

**SVC Parameters - C:** A regularization parameter used to set the tolerance of the model to allow the misclassification of data points in order to achieve lower generalization error. Higher the value of C, lesser is the tolerance. Smaller the value of C, larger is the tolerance of misclassification.

**SVC Parameters - gamma:** The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The lower values of gamma result in models with lower accuracy and the same as the higher values of gamma. It is the intermediate values of gamma which gives a model with good decision boundaries.

**ExtraTrees Parameters - random\_state:** Like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement.

**ExtraTrees Parameters - n\_estimators:** Like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement.

**MLP Parameters - random\_state:** Controls the randomness.

**MLP Parameters - max\_iter:** Maximum number of iterations.

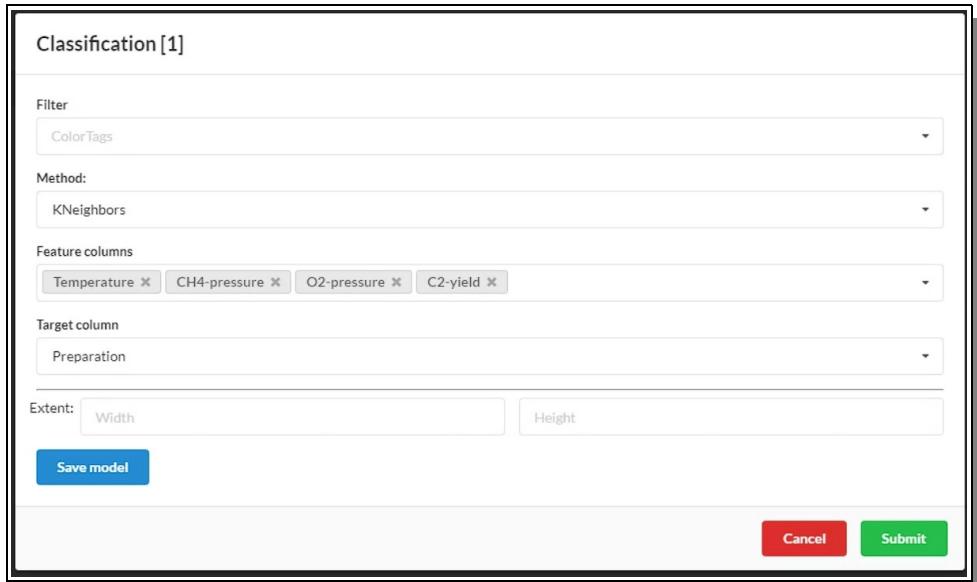
**Ridge Parameters - alpha:** Regularization strength; must be a positive float. Regularization improves the conditioning of the problem and reduces the variance of the estimates. Larger values specify stronger regularization.

In Figure 55 below you can see how the Classification configuration form look and where each field is edited.

How to properly set the various parameters in the configuration form, is a true art form and science all by itself, so that is not something we can tell you, but by testing various settings you might find the

## CADS Component Manual

most optimal one for your data.



The image shows a screenshot of the 'Classification [1]' configuration form. It includes fields for 'Filter' (set to 'ColorTags'), 'Method' (set to 'KNeighbors'), 'Feature columns' (listing 'Temperature', 'CH4-pressure', 'O2-pressure', and 'C2-yield'), 'Target column' (set to 'Preparation'), and 'Extent' (with 'Width' and 'Height' fields). A 'Save model' button is at the bottom left, and 'Cancel' and 'Submit' buttons are at the bottom right.

Figure 55: The Classification CADS component configuration form being edited

When the configuration form is submitted it will contact the server in order to do the calculations requested using the Scikit Learn library. Depending on the data size and the configuration setting this might take some time and a waiting indicator will show up to visualize that, though the exact time will of course not be known in advance. Whether it takes 20 seconds or 20 minutes is a matter of ML experience, but the web-server will time out after a certain time, so keep the data size and settings such so that the time is not becoming extreme.

In Figure 56 below you can see the result of a Kneighbors Classification on some chemical data plotted in a familiar way.

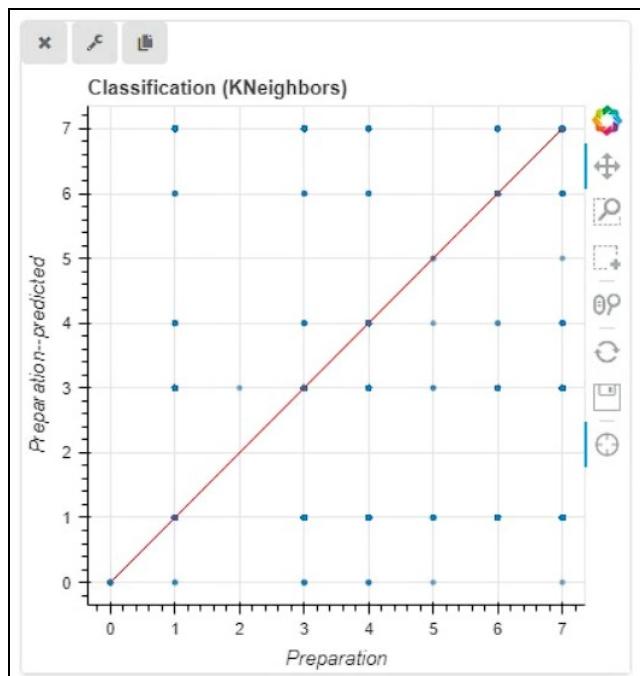


Figure 56: A KNeighbors Classification visualization result in the CADS component

## CADS Component Manual

At the bottom of the configuration form, as can be seen in Figure 55 in the previous page, there is a 'Save Model' button, which is disabled at start, but after the training is done and finished becomes enabled. If one press that button a popup window shows which asks the user to give this model prediction a name, and after doing so and clicked 'Save', the server will once again be contacted and the loading indicator show up. When it is done, the model will then be available for use under the Prediction tab in the top main menu, as detailed explained in the CADS User Manual.

## K-Means Clustering

K-Means Clustering is a method in machine learning where data points are grouped into clusters based on their similarity, with K representing the number of clusters.

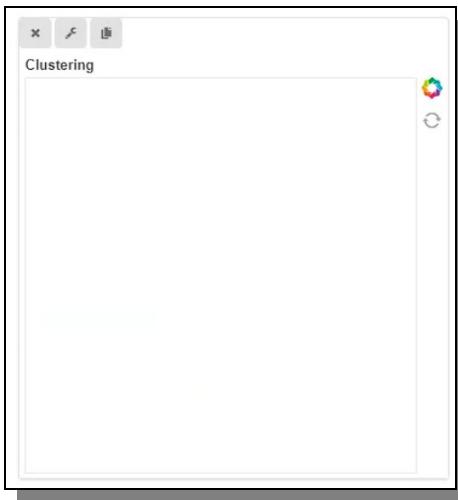


Figure 57: A newly loaded, empty Clustering component ready to be used.

The CADS Clustering component is supported in the back by Scikit Learn and allows for visualization of detected clusters in basically two ways, a classic 2D plot or as a bar chart. The configuration is set to a minimum of options for easy and quick results.

When first loaded, the view is empty, as seen in Figure 57 above, since before showing anything the user need to configure how the Clustering should be done, so the first thing to do is to enter the configuration form and set the available parameters.

The fields available for the Clustering component and which are found in the Clustering configuration form are kept to a minimum for a quick and an easy setup and they are explained more in detail in the lists below.

### ***Clustering Component's Form Fields (global in all settings)***

---

**Visualization Type:** Whether to display the clusters in a common classic 2D scatter plot or just more simple as a Bar Chart.

**Number of Clusters:** The amount of clusters the cluster algorithm should look for in the data.

**Featured Columns:** Which feature columns in the data that should be used when identifying possible clusters

**Extent:** All components have width and height values encompassed in the extent group

### ***Clustering Component's Form Fields (local based on selected settings)***

---

**Scatter Plot - X:** If displaying the clusters in a classic 2D scatter plot then one need to pick which data column should be used for the X-axis.

## CADS Component Manual

**Scatter Plot - Y:** If displaying the clusters in a classic 2D scatter plot then one need to pick which data column should be used for the Y-axis.

**Bar Chart - Method:** If displaying the clusters as a simple bar chart then one has the option to use GMM (Gaussian mixture models) instead of the default main clustering algorithm of K-Means.

In Figure 58 below you can see how the K-Means Clustering configuration form look and where each field is edited.

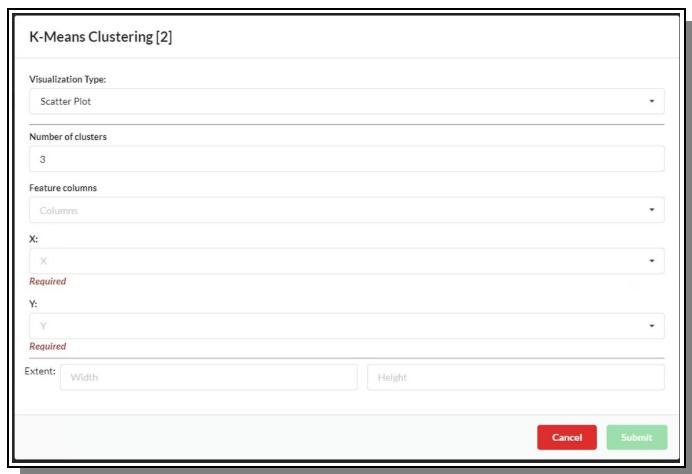


Figure 58: The Clustering CADS component configuration form when setting up a scatter plot.

When the configuration form is submitted it will contact the server in order to do the calculations requested using the Scikit Learn library. Depending on the data size and the configuration setting this might take some time and a waiting indicator will show up to visualize that, though the exact time will of course not be known in advance. If it takes too long there is a risk that the web-server times-out

In Figure 56 below you can see the result of the K-Means Clustering visualized in a scatter plot where each cluster has been assigned its own color.

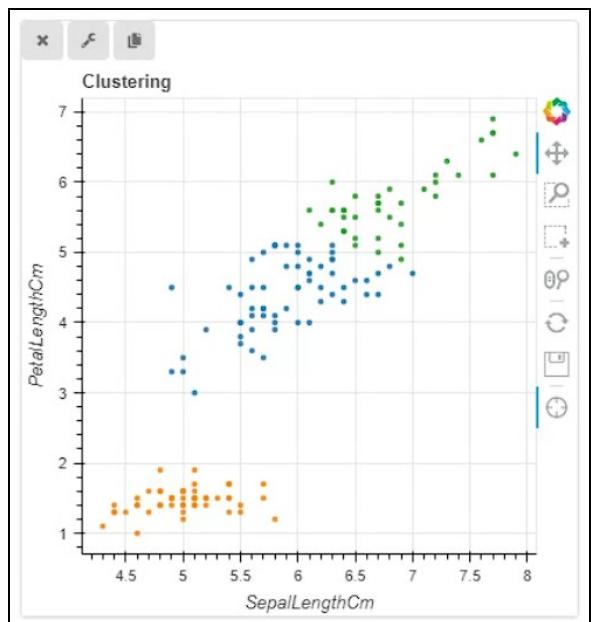


Figure 59: Each cluster gets its own color in the scatter plot of the component.

## CADS Component Manual

If instead the user wish to view the clusters as a bar chart then they would just change the visualization type in the configuration form (and perhaps optionally the clustering method) as can be seen in Figure 60 below.

The screenshot shows the 'K-Means Clustering [2]' configuration form. The 'Visualization Type:' dropdown is set to 'Bar Chart'. The 'Method:' dropdown is set to 'KMeans'. The 'Number of clusters' input field contains the value '3'. The 'Feature columns' dropdown is set to 'Columns'. The 'Extent:' section contains 'Width' and 'Height' input fields. At the bottom right are 'Cancel' and 'Submit' buttons.

Figure 60: The Clustering CADS component configuration form when setting up a bar chart.

When the configuration form is submitted it will as previously contact the server in order to do the calculations requested, also this time using the Scikit Learn library. The process calculations may take some time as mentioned previously.

In Figure 61 below you can see the result of the Clustering visualized in a bar chart where each cluster has been assigned its own bar.

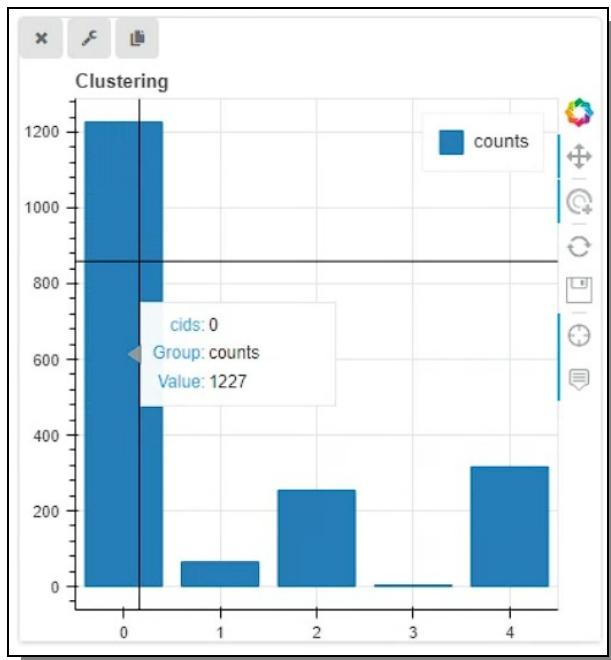


Figure 61: Each cluster gets its own bar and shows how many data points is inside each cluster.

## Gaussian Process

The Gaussian Process Regression is a powerful machine learning technique that allows for predicting values and standard deviations based on user inputs. Furthermore, it can also visualize expected improvement using Bayesian optimization.

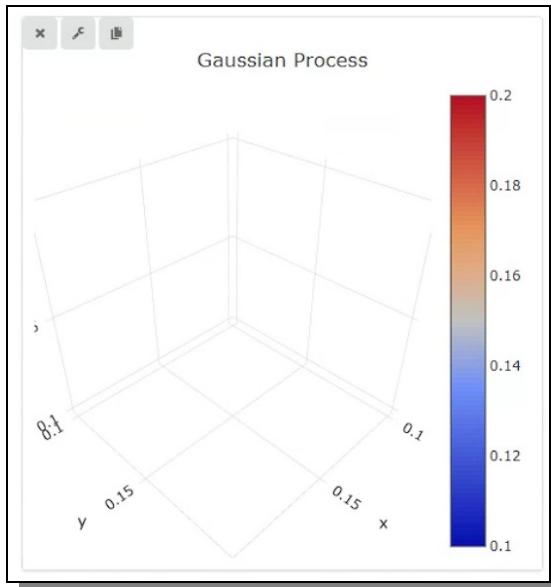


Figure 62: An empty Gaussian Process Component just loaded into CADS Visualization.

The CADS Gaussian Process component is supported in the back by Scikit Learn for the machine learning algorithms and by Plotly 3D scatter plot among other for the front end visualization. The configuration is packed with options for best result and optimal visualization.

When first loaded, the view is empty, as seen in Figure 62 above, since before showing anything the user need to configure how the Gaussian Processing should be done, so the first thing to do is to enter the configuration form and set the available parameters.

The fields available for the Gaussian Processing component and which are found in the Gaussian Processing configuration form are allowing for a wide range of control of the final result and they are explained more in detail in the lists below.

### ***Gaussian Process Component's Form Fields (global in all settings)***

**Feature columns - Number of elements:** The number of learning elements (1-500).

**Feature columns – Add Feature:** Button that generate a set of fields for each Feature column the user wish to add. Add and remove as many as wanted/needed.

**Feature columns – First field:** Select which column you wish to include in the process.

**Feature columns – Second field - Min:** The minimum value to be used in the process.

**Feature columns – Third field - Max:** The maximum value to be used in the process.

**Target Column:** Select the target column for the process

**Target Column – Minimization / Maximization:** Choose whether to update the maximum or minimum value.

**Kernel:** Select which Kernel to use. Try using the kernel with the highest score as explained below.

**Get Score:** After selecting a Kernel, press the 'Get Score' button to find how well it does. Test for each available kernel and use the one with the highest score.

**Information:** This is what type of information and visualization the user wish to get / see.

## CADS Component Manual

There are four to chose from. 'Prediction', 'Standard Deviation' or 'Expected Improvement' will be shown as a plot. Which type of plot depends on the number of features chosen, but for 2 features and one target a 3D plot will be used. 'Proposed Experimental Conditions' generates a table that shows the top 10 percentile of expected improvement and there corresponding feature columns.

**Extent:** All components have width and height values encompassed in the extent group

In Figure 63 below you can see how the K-Means Clustering configuration form look before being edited and where each field is edited.

The screenshot shows a configuration form titled "Gaussian Process [2]". The "Feature Columns" section contains a "Number of elements" input field with the value "100" and "Add Feature" and "Remove Feature" buttons. A red error message "At least one feature columns must be entered" is displayed. The "Target Column" section has a dropdown menu set to "target column" with a "Required" label. The "Kernel" section has a dropdown menu set to "Kernel" with a "Required" label. The "Information" section has a dropdown menu set to "information" with a "Required" label. The "Extent" section has two input fields both set to "450". At the bottom are "Cancel" and "Submit" buttons.

Figure 63: An empty, not yet edited, Gaussian Process configuration form ready to be used.

And in Figure 64 below you can see a Gaussian Process configuration form while it is being edited and setup, especially focusing on which type of kernels are available. After the user have selected feature columns, target column and the type of kernel, they can then examine the score of the kernel by clicking 'Get Score' button as seen in Figure 65 on the next page.

When all configuration is set and the wanted information / visualization is selected the user may press the 'Submit' button. The data and its settings are then sent to the server for calculation, which may take some time, as the loading indicator will show, if the dataset is large, but usually will be finished in just a few seconds.. After all calculations are done the Gaussian Process component will be updated accordingly and displayed as requested.

## CADS Component Manual

Gaussian Process [3]

Feature Columns:  
Number of elements  
400

Si/AI2 min 0 max 20  
Cu IER(%) min 0 max 100

Add Feature Remove Feature

Target Column  
TON  
Maximization

Kernel  
ConstantKernel() \* RBF(np.ones()) + WhiteKernel() + ConstantKernel() \* DotProduct()  
ConstantKernel() \* RBF() + WhiteKernel()  
ConstantKernel() \* DotProduct() + WhiteKernel()  
ConstantKernel() \* RBF() + WhiteKernel() + ConstantKernel() \* DotProduct()  
ConstantKernel() \* RBF(np.ones()) + WhiteKernel()  
ConstantKernel() \* RBF(np.ones()) + WhiteKernel() + ConstantKernel() \* DotProduct()

Extent: 377 350

Cancel Submit

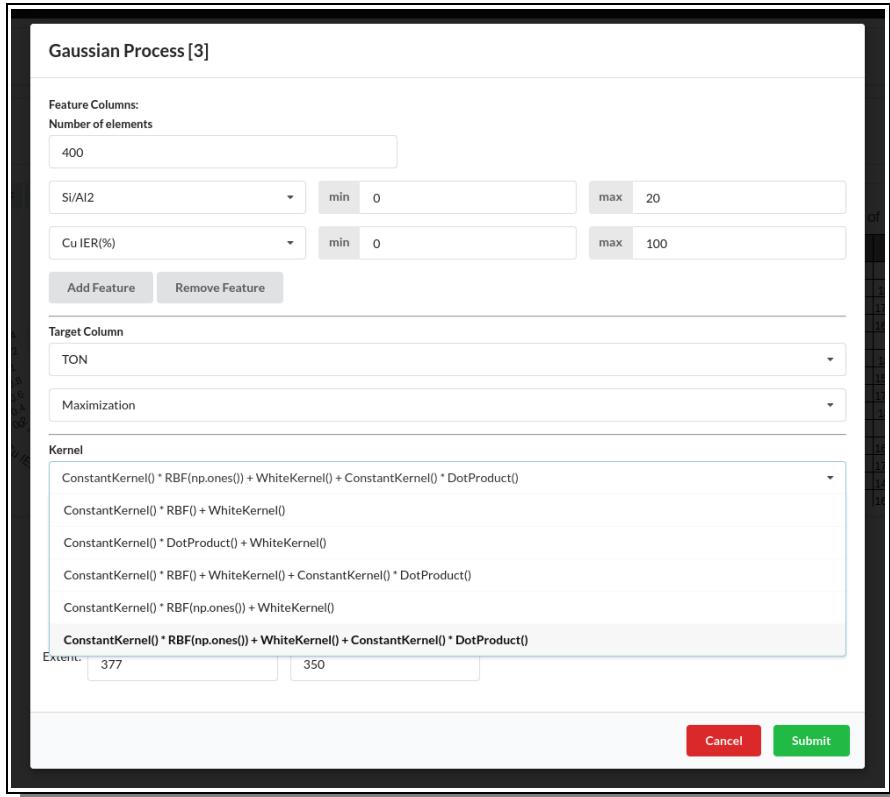


Figure 64: In the process of editing the Gaussian Process configuration form.

Gaussian Process [3]

Feature Columns:  
Number of elements  
400

Si/AI2 min 0 max 20  
Cu IER(%) min 0 max 100

Add Feature Remove Feature

Target Column  
TON  
Maximization

Kernel  
ConstantKernel() \* RBF(np.ones()) + WhiteKernel() + ConstantKernel() \* DotProduct()

Get Score  
No Info

Information:  
Prediction

Extent: 377 350

Cancel Submit

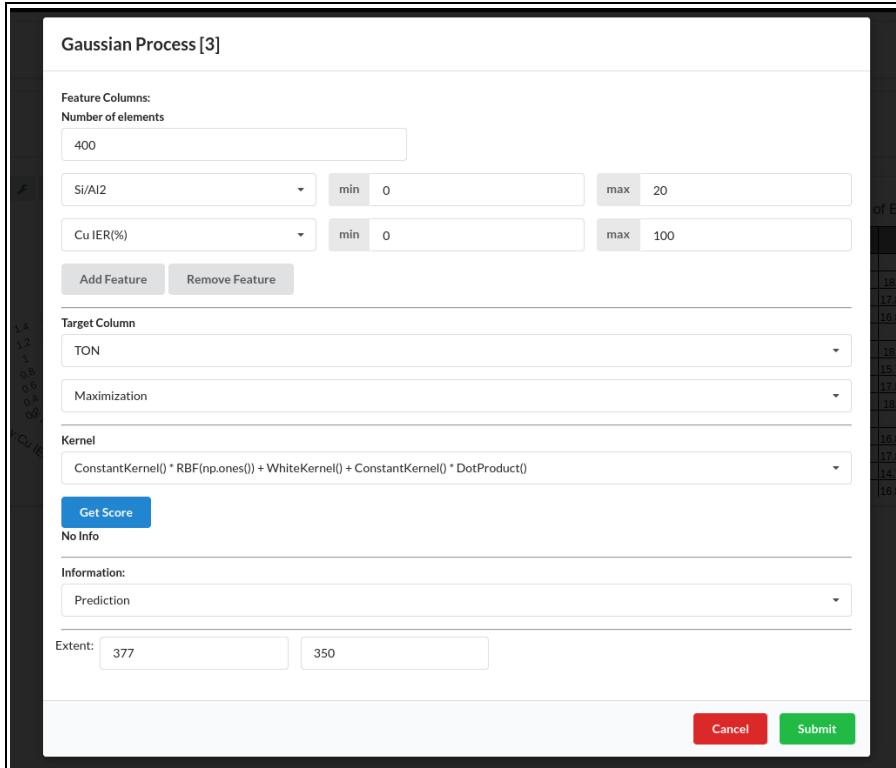


Figure 65: The Gaussian Process configuration form after all fields have been set.

## CADS Component Manual

In Figure 66 below one can see a set examples of how the Gaussian Process component data is visualizing its results.

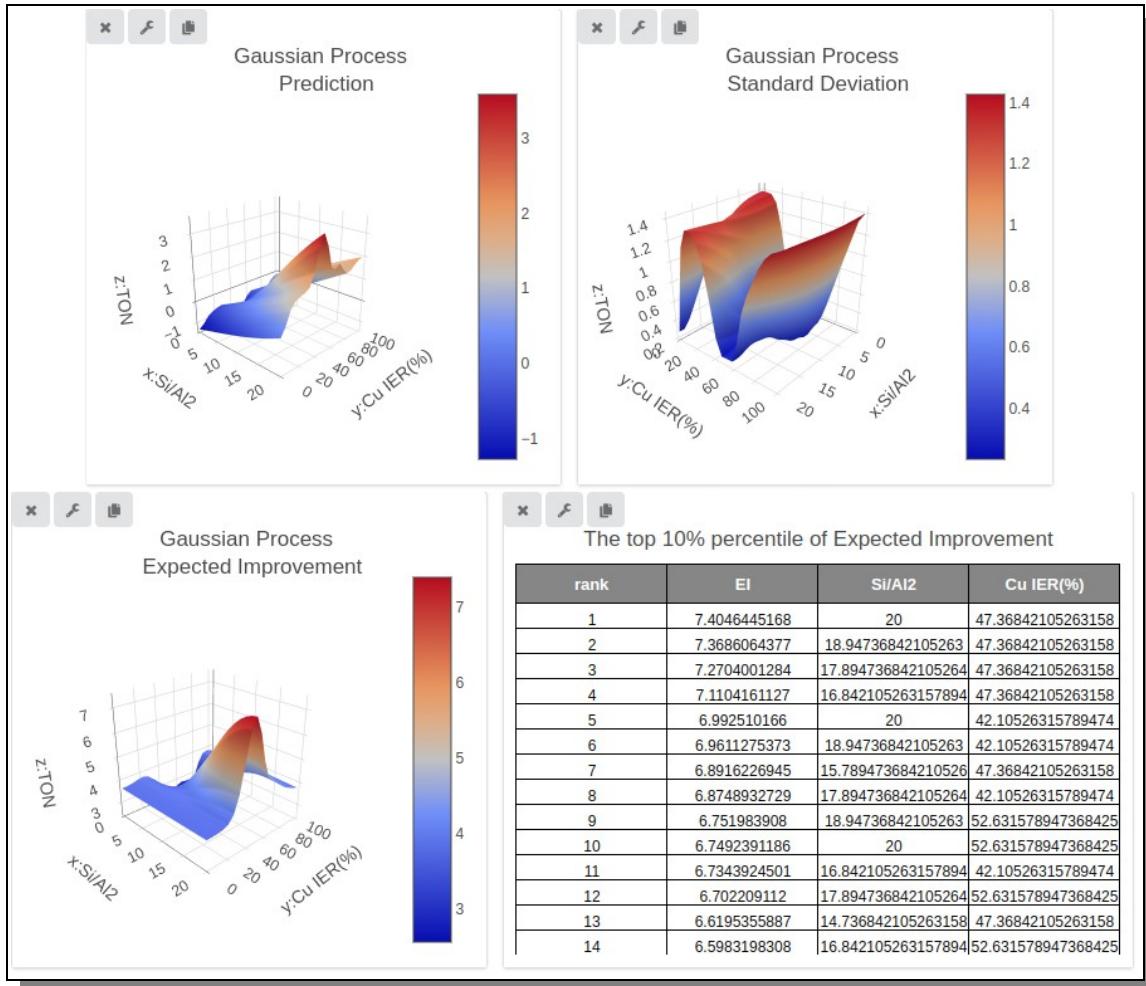


Figure 66: Predictions of Gaussian Process regression and bayesian optimization

## Statistics

The CADS Statistics component is just a convenient way to collect, store and display some useful data information.

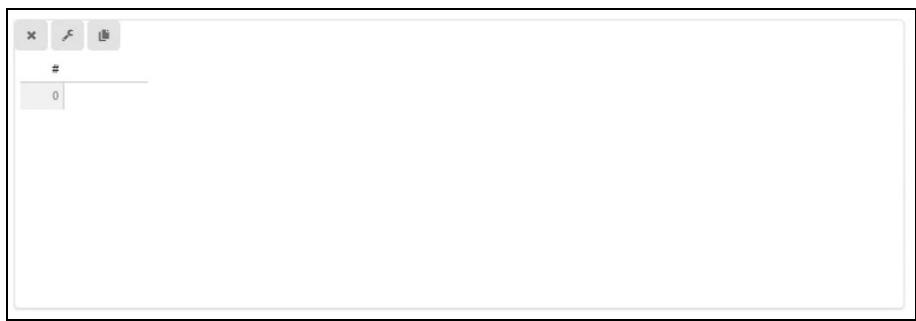


Figure 67: An empty Statistics component not yet fully connected to the current data.

## CADS Component Manual

When first loaded the view is empty, as seen in Figure 67 above, since before showing anything the user need to configure what parts of the data should be involved, so the first thing to do is to enter the configuration form and set the available parameters.

The fields available for the Statistics component are ridiculously few and should be finished in a second but still explained more in detail in the list below.

### ***Classification Component's Form Fields (global in all settings)***

**Featured Columns:** Which feature columns in the data that should be used when calculating the data statistics.

**Extent:** All components have width and height values encompassed in the extent group

In Figure 68 below you can see how the Statistics configuration form look and where each field is edited.

Statistics [2]

Feature columns

SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

Extent: 800 230

Cancel Submit

Figure 68: The Statistics Configuration form is just about selecting which columns to include.

The statistics that will be calculated and displayed for the data is as follows; 'count' that just tells how many there are, 'mean' which is the mean value for the data in the specified columns, 'std' is the standard deviation value for the data in the specified columns, 'min' is the minimum value for the data in the specified columns, '25%' is the 25% point value for the data in the specified columns, '50%' is the 50% point value for the data in the specified columns, '75%' is the 75% point value for the data in the specified columns and , 'max' is the maximum value for the data in the specified columns.

In Figure 69 below you can see the result of a statistics calculation on the classic Iris dataset.

#	Stats	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	count	150	150	150	150
1	mean	5.84333	3.054	3.75867	1.19867
2	std	0.82807	0.43359	1.76442	0.76316
3	min	4.3	2	1	0.1
4	25%	5.1	2.8	1.6	0.3
5	50%	5.8	3	4.35	1.3
6	75%	6.4	3.3	5.1	1.8
7	max	7.9	4.4	6.9	2.5

Figure 69: The Statistics component showing some statistics for the current dataset.

## CADS Component Manual

### Static Data Visual Support Components

The Static Visual Support components are components that does not really interact with the workspace loaded data, but instead do its own thing with its own internal data or data collected elsewhere but still is showing things that might be useful to look at in parallel at the side or in addition. Here we find components like a Periodic Table and a Molecule Viewer.

### Periodic Table

The CADS Periodic Table component is exactly what it sound like, a periodic table containing all usual data a periodic table contains.

In Figure 70 below one can see how it looks when loaded into the CADS website.

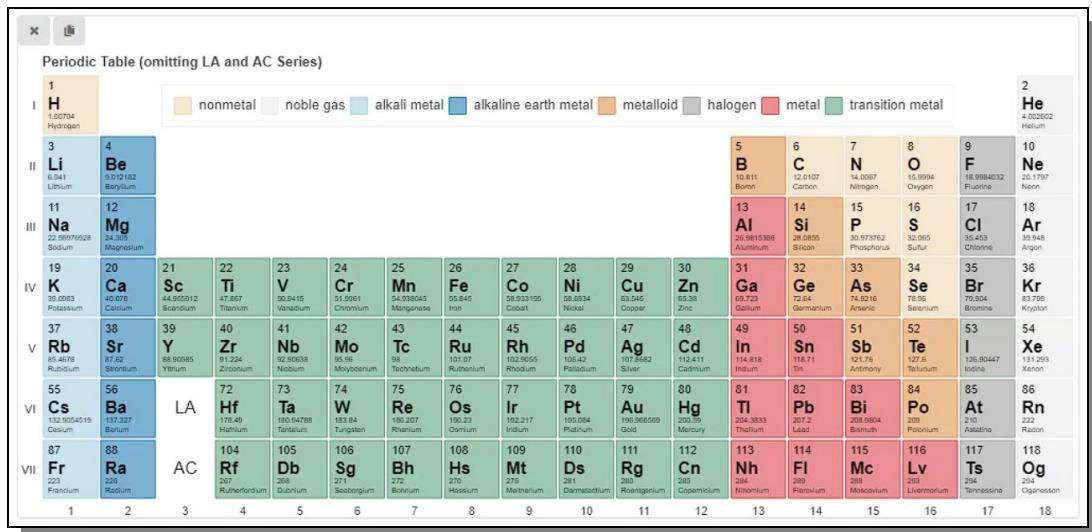


Figure 70: A classical periodic table represented by a CADS component.

This component does not care nor connect with the dataset that is currently loaded, and is more for visualization purposes.

The component is slightly interactive though, like most other CADS components, and one thing available is the power to select specific elements in order to highlight them and which can be seen in Figure 71 below.

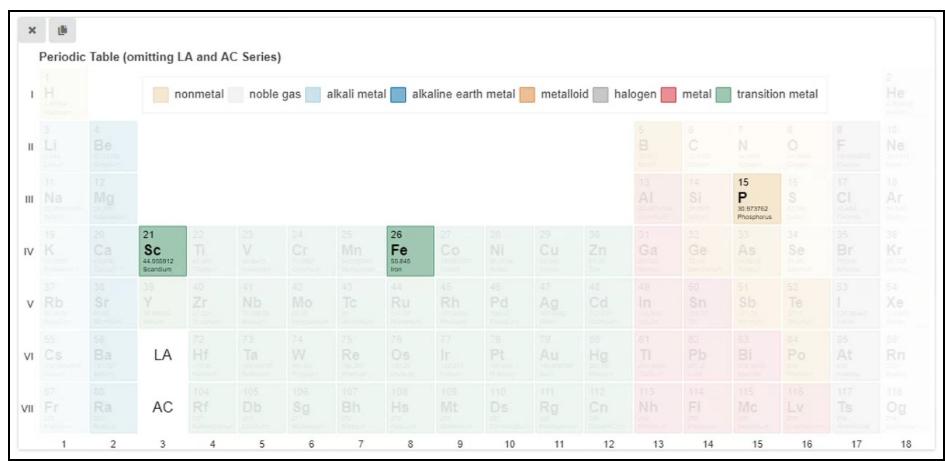


Figure 71: Some elements selected and highlighted in the periodic table.

## CADS Component Manual

### Molecule 3D

The CADS Molecule 3D component is not connected to the dataset currently loaded in the Analysis workspace. Instead it is used for loading most of the commonly used file formats when displaying 3D atomic models, such as mol, xyz and cif files.

When the component is first loaded into the workspace it is empty and look like shown in Figure 72 below.

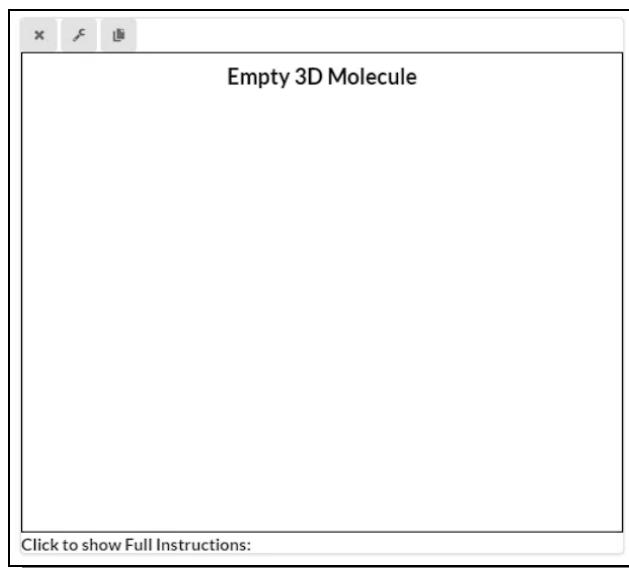


Figure 72: An empty, newly loaded, Molecule 3D component.

In order to get a molecule to view and interact with the user needs to open the configuration form and add the required data. In Figure 73 below you can see how that form looks like and with fields it contains.

A screenshot of the "Molecule3D [1]" configuration form. It includes fields for Background Color (with a preview swatch), Text Color (with a preview swatch), and a "Load File" button for MOL, XYZ or CIF files. Below the file input is a text area containing chemical data. Fields for Chemical Name (Propane) and Chemical Formula (C3H8) are present, along with a URL for further details (http://www.chemspider.com/Chemical-Structure.6094.html). At the bottom, there are "Extent" settings (500, 400) and "Cancel" and "Submit" buttons.

Figure 73: The Molecule 3D configuration form in all its glory.

## CADS Component Manual

The fields available for the Molecule 3D component not difficult to understand but we will still explain them in more details in the list below.

### ***Molecule 3D Component's Form Fields (global in all settings)***

**Background Color:** The background color of the molecule view port. (Default is white)

**Text Color:** The text color for all text inside the molecule view port. (Default is black)

**Load File:** A button that allows the user to select a file on the local computer in one of the allowed file formats (mol, xyz, cif) and load into the component.

**Chemical Name:** If the chemical name value can be found inside the file, then it will be extracted and inserted here, otherwise the user can edit this as free text.

**Chemical Formula:** If the chemical formula value can be found inside the file, then it will be extracted and inserted here, otherwise the user can edit this as free text.

**Further Details URL:** If there is a URL value found inside the file, then it will be extracted and inserted here, otherwise the user can edit this freely.

**Extent:** All components have width and height values encompassed in the extent group

The configuration form also have a link to the popular ChemSpider website where one can find a wide range of molecule files for download if the user does not yet have any of their own, or is looking for some particular ones. One can also use Google to find these kind of files.

In Figure 74 below you can how the component looks like after a molecule file has been loaded and the form have been submitted.

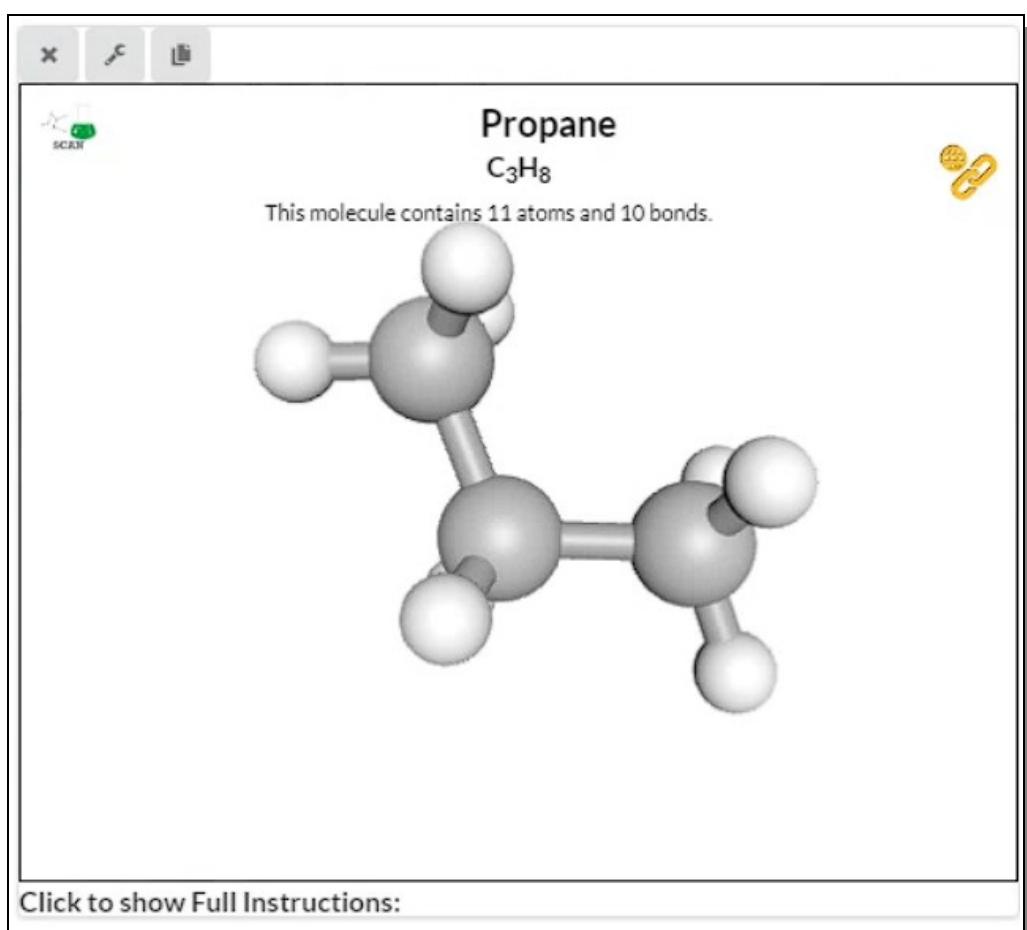


Figure 74: A molecule file loaded into the component for interaction and use.

## CADS Component Manual

At the top left corner there is a link to another online tool developed by this research group, and for some this link might be useful, but for most people it can be safely ignored.

At the top right corner there is a link to ChemSpider for the currently loaded molecule if it is available there.

At the bottom beneath the molecule view port there is list of instructions for all available interaction features this component is offering. By clicking on it one can toggle it to be displayed or hidden.

In Figure 75 below one can see the same molecule displayed as before but now some of the interaction feature settings have been changed.

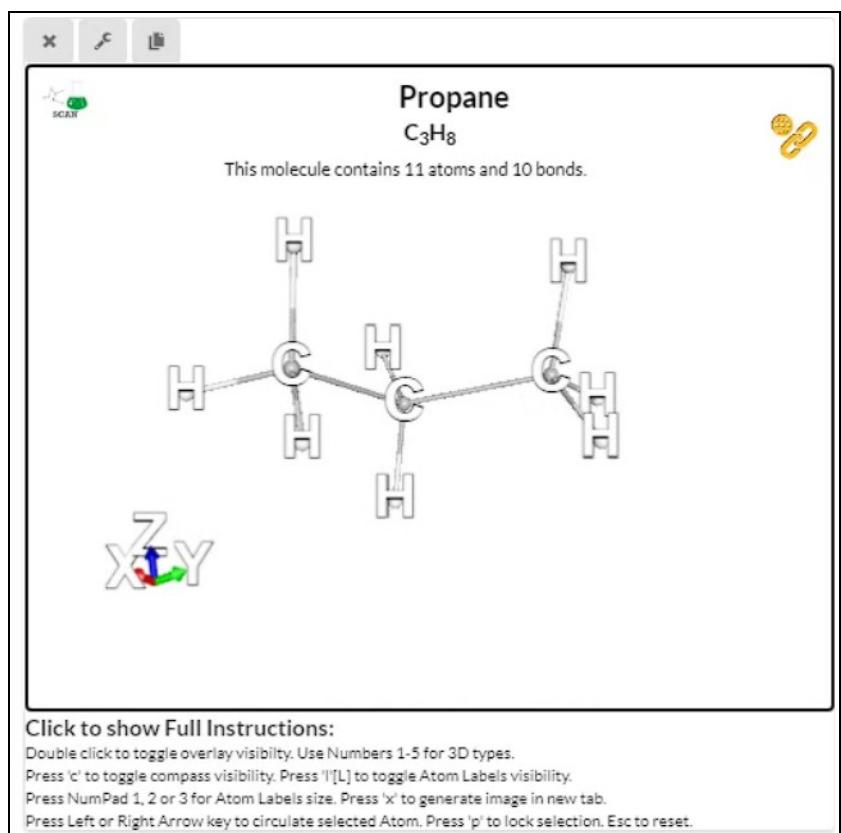


Figure 75: Same molecule but displayed in another way and also showing the interaction instruction available for this component.

Let's explain more in detail what these features are.

If one double clicks in the molecule view port, one toggles the visibility of the overlay text, such as molecule name and more.

Selecting a number 1-5 on the keyboard while the mouse is above the view port switches which drawing style should be used for the molecule. (1 is default drawing style)

Press 'c' on the keyboard for toggling the visibility of a compass showing the direction of the X, Y and Z axis in the 3D space. (default is off)

Press 'l' (L) on the keyboard for toggling the visibility of the atomic name labels of all atoms. (default is off)

Press NumPad '1', '2' or '3' for selecting the size of the atom name labels, going from small, medium to big. (medium is default)

## CADS Component Manual

Press 'x' to generate an image of the molecule and open it in a new tab in the browser. To save the image to your local computer the user need to do what one usually does to save an image from the web browser.

Press the left or right arrow key on the keyboard to circulate among the displayed atoms and make it selected. A selected atom becomes green and big to clearly indicate its selection status.

Press 'p' on the keyboard to lock a selection, before moving on to select another. A locked selected selected atom becomes yellow and bigger to clearly indicate its locked selection status as can be seen in Figure 76 below.

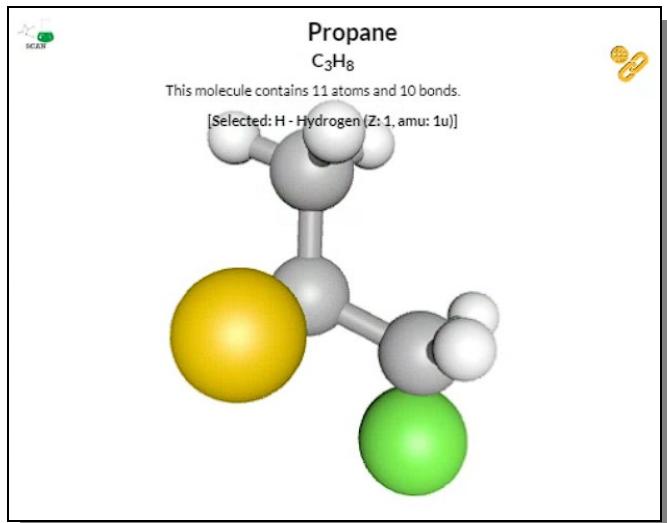


Figure 76: An interactive 3D molecule with some atoms selected.

When a second atom is locked, all selections reset and disappear, and instead the atomic distance between the two selected atoms is displayed as can be seen in Figure 77 below. If this is repeated for another pair of atoms the previous distance value is removed.

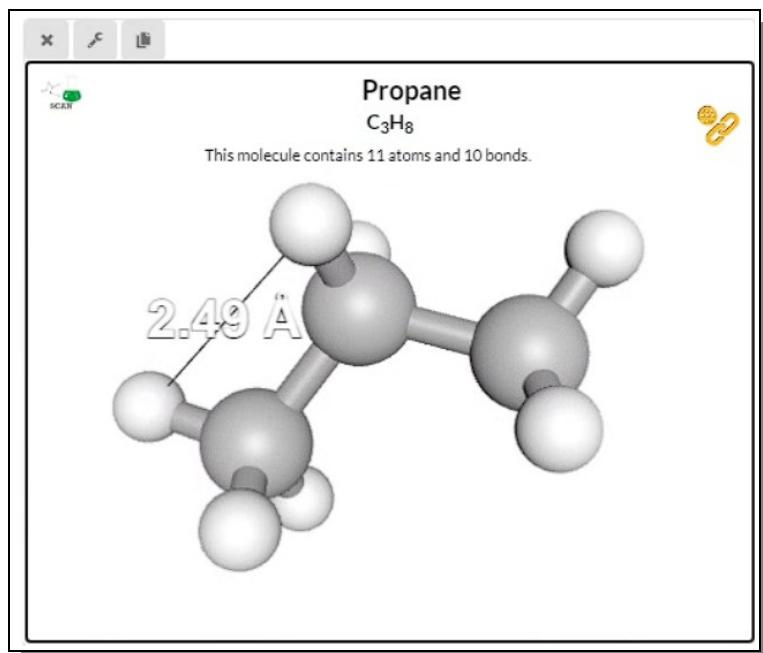


Figure 77: Showing the atomic measurement distance between two selected atoms.

## CADS Component Manual

### **Summary**

We wish that this manual have told you everything you need to know about all currently available CADS Analysis-components. Feel free to return here whenever your memory fails you on what to do and when.

Also, please read our other detailed manuals on how to use CADS in general terms, our manual on how to setup and maintain your own CADS server and/or our development manual on how to code and develop new CADS components.

If you ever come up with some ideas about a possible future component, please do not hesitate to contact us.

Congratulations on your newly gained status as an ‘Advanced CADS-User’. Now it is time to do some serious analyzing research. Enjoy!



# Index

- Analysis.....2, 3, 5, 14, 27, 33, 55, 59  
 Axes.....27, 28  
 Bar.....6, 10, 30, 46, 47, 48  
 Bins.....13, 29  
 Bokeh.....4, 5, 6, 8, 10, 13, 15  
 Chan-Vese Segmentation.....25  
 Chemical.....56  
 Classification.....40, 43, 44, 45, 53  
 Clustering.....46, 47, 48, 50  
 Color7, 8, 10, 13, 14, 17, 18, 19, 20, 21, 24, 25, 26, 28, 31, 32, 34, 35, 44, 47, 56  
 Column.7, 8, 10, 13, 14, 16, 17, 27, 28, 29, 30, 31, 32, 34, 35, 41, 44, 46, 47, 49, 50, 53  
 Component 2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 15, 16, 17, 18, 19, 20, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 44, 46, 49, 50, 52, 53, 54, 55, 56, 57, 59  
 Contour.....26  
 Contrast.....20, 25, 32  
 Convex Hull.....25  
 CSS.....20  
 Data Processing.....18  
 Denoise.....25  
 Dimensions.....10, 14  
 DragCoeff.....35  
 Edge Detection.....25  
 Extent7, 8, 10, 13, 15, 17, 20, 28, 29, 30, 31, 32, 35, 41, 44, 46, 50, 53, 56  
 ExtraTrees.....42, 44  
 Feature importance.....30  
 Featured Columns.....30, 32, 41, 44, 46, 53  
 Filter.....7, 8, 10, 20, 21, 25, 28, 44  
 Flip.....25  
 Fluorescent.....26  
 Formula.....56  
 Gamma.....25, 41, 44  
 Gaussian Process.....49, 50, 52  
 Gray scale.....20, 24  
 Heat map.....31, 32  
 Histogram.....27, 29  
 HOG.....26  
 HSV.....25  
 Image.3, 4, 6, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 35, 58  
 ImageView.....18, 19, 20, 24  
 K-Means.....46, 47, 50  
 KernelRidge.....42  
 KFold.....41, 42  
 Line.....16, 17  
 Link.....5, 34, 35, 38, 56, 57  
 Machine learning.....3, 40, 43, 46, 49  
 Measures.....10  
 Method.10, 14, 25, 27, 32, 41, 42, 44, 46, 47, 48  
 MLP.....42, 44  
 Molecule 3D.....55  
 Network.....33, 34, 35, 36, 37, 38, 39  
 Node.....33, 34, 35, 37, 39, 40, 41, 44  
 NodeGraph.....33, 34  
 Pairwise correlation.....32, 33  
 Palette.....10, 13, 14, 17, 31, 32  
 Parallel Coordinates.....27, 28  
 Periodic Table.....54  
 Pie.....6, 13  
 Plot.....3, 4, 6, 8, 15, 16, 43, 45, 46, 47, 49, 50  
 Plotly.....4, 6, 15, 16, 49  
 RAG.....25  
 RandomForest.....41, 44  
 Regional Maxima.....25  
 Regression.....40, 41, 42, 43, 44, 49  
 Ridge detection.....25  
 Rotate.....14, 20, 24  
 Save Model.....43, 46  
 Scatter.....3, 8, 14, 46, 47, 49  
 Scatter 3D.....14  
 Scikit....4, 18, 19, 20, 21, 22, 23, 24, 42, 45, 46, 47, 48, 49  
 Sharpen.....25  
 Skeletonize.....26  
 Source.....34  
 SpringCoeff.....35  
 Static Data Visual Support.....54  
 Statistics.....52  
 Swirl.....25  
 Table.....1, 6, 7, 16, 50, 54  
 Target...3, 13, 14, 25, 29, 30, 34, 40, 41, 44, 49, 50  
 Target column.....13, 14, 30, 41, 44, 49, 50  
 Thresholding.....25  
 TrainTestSplit.....41, 42  
 Transpose.....10

## **CADS Component Manual**

- Visualization. 2, 3, 6, 27, 31, 32, 46, 48, 49, 50,  
54  
VivaGraph.....6, 38
- X.....8, 14, 17, 31, 46, 57  
Y.....8, 10, 14, 17, 31, 47, 57

# Figure Index

Figure 1: A classic table showing the data in rows and columns.....	6
Figure 2: The Table Form is as simple as it gets.....	7
Figure 3: A formatted table with selected columns and selected rows.....	7
Figure 4: One can filter which rows to be seen and select which rows to be visible.....	8
Figure 5: A modified Table component for a more specific look and feel.....	8
Figure 6: The Scatter plot components displaying Iris data.....	9
Figure 7: The Scatter plot form configured for a particular view.....	9
Figure 8: A classic bar chart showing multiple data points.....	11
Figure 9: The Bar Chart form default view configured to display the data in shown specific way..	11
Figure 10: Transposed data displayed in the Bar Chart component.....	12
Figure 11: The Bar component easily lets the user transpose the data for viewing in new angles....	12
Figure 12: A classic Pie Chart displaying its data.....	13
Figure 13: The Pie Chart Component configuration Form for easy and quick views.....	13
Figure 14: The Scatter 3D allows us to interact with the data in three dimensions.....	14
Figure 15: The Scatter 3D configuration form lets us setup the view in a quick and easy way.....	15
Figure 16: The Scatter 3D component uses Plotly and therefore have a slightly different set of interaction buttons attached to it.....	15
Figure 17: A classsic Line Chart for all needs that requires it.....	16
Figure 18: It could probably not be any easier than this to configure the Line chart component.....	17
Figure 19: An empty ImageView component that has just been loaded into the workspace.....	18
Figure 20: The ImageView configuration form as it looks before any changes have yet been done.	19
Figure 21: A simple image loaded from the local machine into the component.....	20
Figure 22: The ImageView component form focusing on CSS image filters that will alter the image.	21
Figure 23: The ImageView Configuration form focusing on the SciKit-Image processing availability.....	22
Figure 24: An example of some image manipulation done to an image using SciKit-Image.....	23
Figure 25: The available SciKit-Image processes that can be evoked to alter and enhance an image.	23
Figure 26: A detailed view on all available parameters for all available SciKit-Image processing one can apply.....	24
Figure 27: Here the annotation capabilities have been enabled and used for image enhancement....	26
Figure 28: A CADS Parallel Coordinate component displaying the data for a set of columns.....	27
Figure 29: The parallel coordinate configuration form is very slim and easy to use.....	27
Figure 30: Some parts of some axis have been selected in order to focus on a particular part of the data.....	28
Figure 31: Here some of the data has been selected and colored as described above.....	28
Figure 32: One histogram component displaying four histograms for four data columns.....	29
Figure 33: The Histogram configuration form could probably not be any simpler.....	29
Figure 34: Displaying the featured importance for the selected columns in the data.....	30
Figure 35: The Featured Importance configuration form is easy and straight forward.....	30
Figure 36: A CADS Heatmap component displaying data the heatmap way.....	31
Figure 37: The Heat Map Configuration form should not be any problem to edit for a good result.	31
Figure 38: A Pairwise Correlation visualization is also available in CADS.....	32
Figure 39: The pairwise correlation configuration form is as expected simple and easy.....	33

## CADS Component Manual

Figure 40: A simple sample network node graph in CADS.....	33
Figure 41: The Nodes evenly placed after they have been moved by the user.....	34
Figure 42: The Node Graph Configuration Form and all its available settings to control the network. .....	36
Figure 43: A proper node graph drawn based on real network data.....	36
Figure 44: A network graph using a bunch of the available settings in the configuration form, such as e.g. bezier curve and color and size gradiance.....	37
Figure 45: Displaying the names for each node in the network graph.....	37
Figure 46: Displaying the names for every link between nodes in the network graph.....	38
Figure 47: A really large node network graph loaded into the component with thousands of nodes.	38
Figure 48: A node and its neighbors have been selected.....	39
Figure 49: Multiple select of many nodes is also possible inside the component.....	39
Figure 50: Calculating and showing the shortest path between two nodes in the graph.....	40
Figure 51: A newly loaded, empty Regression component ready to be used.....	41
Figure 52: The Regression CADS component configuration form being edited.....	42
Figure 53: A Random Forest Regression visualization result in the CADS component.....	43
Figure 54: A newly loaded, empty Classification component ready to be used.....	43
Figure 55: The Classification CADS component configuration form being edited.....	45
Figure 56: A KNeighbors Classification visualization result in the CADS component.....	45
Figure 57: A newly loaded, empty Clustering component ready to be used.....	46
Figure 58: The Clustering CADS component configuration form when setting up a scatter plot.....	47
Figure 59: Each cluster gets its own color in the scatter plot of the component.....	47
Figure 60: The Clustering CADS component configuration form when setting up a bar chart.....	48
Figure 61: Each cluster gets its own bar and shows how many data points is inside each cluster.....	48
Figure 62: An empty Gaussian Process Component just loaded into CADS Visualization.....	49
Figure 63: An empty, not yet edited, Gaussian Process configuration form ready to be used.....	50
Figure 64: In the process of editing the Gaussian Process configuration form.....	51
Figure 65: The Gaussian Process configuration form after all fields have been set.....	51
Figure 66: Predictions of Gaussian Process regression and bayesian optimization.....	52
Figure 67: An empty Statistics component not yet fully connected to the current data.....	52
Figure 68: The Statistics Configuration form is just about selecting which columns to include.....	53
Figure 69: The Statistics component showing some statistics for the current dataset.....	53
Figure 70: A classical periodic table represented by a CADS component.....	54
Figure 71: Some elements selected and highlighted in the periodic table.....	54
Figure 72: An empty, newly loaded, Molecule 3D component.....	55
Figure 73: The Molecule 3D configuration form in all its glory.....	55
Figure 74: A molecule file loaded into the component for interaction and use.....	56
Figure 75: Same molecule but displayed in another way and also showing the interaction instruction available for this component.....	57
Figure 76: An interactive 3D molecule with some atoms selected.....	58
Figure 77: Showing the atomic measurement distance between two selected atoms.....	58

# Thanks for your support



Screenshot of the CADS software interface showing various data analysis and visualization tools:

- Top Bar:** CADS, Data Management, Analysis, Prediction, Docs, Log out, Setting, Save.
- Left Sidebar:** ChemCatChem dropdown, Assign Color, Enable Mobility.
- Release Candidate [v.0.90]:** A table with columns: #, Supp, Cads, Cat0, Cat1, Cat2, Supp, CH4, Supp, C2H, CO, C2H/C2H, CH4, Index, Aniol, Tanol, Aniol, Aniol, Prep, Aniol, Coni, Coni, C2H, Cat0, Cat1, Cat2, C. The table contains 15 rows of data.
- Regression (TrainTestSplit) [ExtraTrees]:** A scatter plot comparing C2yield-Predicted (x-axis, 0 to 30) and C2yield-True (y-axis, 0 to 30). It includes a regression line and points colored by category (blue for Train, red for Test). CV scores: mean r=0.2623766146620979, mean MAE=-4.127841368776916.
- Beta [v.0.80]:**
  - Scatter:** A scatter plot of C2yield vs Preparation (0 to 7). The data shows vertical clusters of points for each preparation level.
  - Pie Chart:** A pie chart titled "The composition of 7 categories in the column of 'Temperature'". The categories and their percentages are: 0.00 - 196.14 (blue), 196.14 - 392.29 (light blue), 392.29 - 588.43 (medium blue), 588.43 - 784.57 (light orange), 784.57 - 980.71 (orange), 980.71 - 1176.86 (dark orange), and 1176.86 - 1373.00 (yellow).