innovative science • intuitive software

# Cresset Nodes for KNIME V2.8.0 - Release Notes

# About Cresset Nodes for KNIME

There are currently 20 nodes available from Cresset that incorporate key Cresset applications into the KNIME environment. Each node wraps the corresponding application making it available within KNIME.

Additional 17 deprecated Forge nodes are still available in this release to support the update of existing workflow to use the new Flare nodes with equivalent functionality.

## Flare™

Flare is a comprehensive drug design platform including both structure-based and ligand-based methods. All Flare KNIME nodes operate the same pyflare command line binary, providing access to all Flare functionality through the Flare Python API.

### Flare Viewer

The Flare Viewer node is used to launch Flare for viewing ligands, proteins and results.

### pyflare

The pyflare node gives access to the Flare Python API from within KNIME. This enables access to all Flare functionality including protein preparation, docking and Electrostatic Complementarity™.

The full documentation for the Flare Python API can be found in the Flare installation directory.

### Flare Align

The Flare Align node gives access to Cresset's unique ligand alignment method. It uses one or more reference molecule(s), supplied in a specified 3D conformation, as a template for the alignment of congeneric or diverse ligands. When used on a congeneric series the tool can help in library design and give a rationale for the prioritization of compounds for synthesis. Using Flare Align on a diverse set of active molecules can help define the requirements of the protein of interest, aiding the synthetic chemist in the design of new actives. Additionally the scores from the alignment routine can be used to select molecules for screening in the same way as using the dedicated Blaze™ virtual screening platform (albeit with a lower throughput). Optionally the Flare Align node can load a protein structure that can be used as an excluded volume for the alignment.

## Flare Docking

The Flare Docking node runs docking and scoring experiments using the Lead Finder™ docking algorithm, now fully integrated in Flare. This docking engine combines a genetic algorithm search with local optimization procedures, which make it efficient in coarse sampling of ligand poses and, following refinement, in providing promising docking solutions.

Lead Finder generates docked ligand poses starting from the 3D structure of an accurately prepared protein structure (Flare can be used to prepare your protein) and one or more 3D ligand structures. Lead Finder assumes that the protein is rigid and analyses the possible conformations of the ligand by rotating functional groups along each freely rotatable bond.

Three different scoring functions are employed, optimized for the accurate prediction of 3D docked ligand poses (LF Rank Score), protein-ligand binding energy (LF dG) and rank-ordering of active and inactive compounds in virtual screening experiments (LF VSscore).

The Flare Docking node supports covalent docking for predicting the binding pose and interactions of covalent inhibitors, in a workflow similar to that of standard docking, but which enables to specify a residue which will covalently bind the docked ligands. The ligands to be docked must be drawn to include an appropriate bond-forming functional group, also called a covalent warhead. Please refer to the Flare manual for a list of the currently supported covalent warheads.

For those docking experiments where the pose of a 'template' ligand is known, this information can be used to bias the docking results for congeneric compounds. When using template docking, the molecules to be docked are aligned by substructure to the template ligand, and the aligned conformation is used to seed the docking run, generally leading to improved docking results. Note that this option is not available for covalent docking.

## 'Models' group

This group includes KNIME nodes for building qualitative and Quantitative Structure-Activity Relationships (QSAR) models in Flare, and scoring new molecules against (Q)SAR models.

### Flare Build Field QSAR

Generates a Field QSAR (3D-QSAR) model for activity from a set of aligned molecules.  The aligned set of molecules is used to derive a set of 'sample positions' around the molecules based on their field points, which can be used to probe any molecule for the electrostatic potential or for the volume taken up at those positions. The data matrix derived from the sample values is then processed by partial least squares (PLS) to generate a linear model for activity. Optionally the data matrix can be exported into the KNIME workspace for use with alternative data analysis methods.

### Flare Score Field QSAR

Calculates predicted activities for molecules against a Field QSAR model. The molecules must be pre-aligned in a manner consistent with the alignment used to create the original Field QSAR model.

### Flare Build Activity Atlas

Generates an Activity Atlas™ model for activity from a set of aligned molecules. Activity Atlas is a method for summarizing the SAR for a ligand series into a visual 3D model that can be used to inform new molecule design, particularly useful for those project teams where there is not enough SAR for a traditional 3D-QSAR approach.

Activity Atlas calculates and displays as 3D visualizations:

- Average of Actives: shows you what the average active molecule looks like, by making an analysis of what the active molecules in the data set have in common.
- Activity Cliff Summary: shows you the critical regions of the SAR, based on activity cliffs.
- Regions Explored analysis: makes an assessment of what regions of the aligned molecules have been fully explored and calculates a novelty score for each molecule in the data set. The Regions Explored analysis can be used within the ‘Flare Score Activity Atlas’ node to compute a 'Novelty' score for new molecules.

The result is a global picture of activity, highlighting those regions around the series where the SAR is clear.

### Flare Score Activity Atlas

Calculates a novelty score for molecules against an Activity Atlas model. The molecules must be pre-aligned in a manner consistent with the alignment used to create the original Activity Atlas model.

### Flare Build kNN

Generates a kNN regression or classification models from a set of molecules.

The kNN methodology is a well-known and robust distance learning approach where the activity for each new compound is predicted as the weighted average activity of its k nearest neighbors in the training set.

The similarity between the molecules is calculated using Cresset's field/shape similarity method or by using the 2D circular fingerprint methods ECFP4, ECFP6, FCFP4, or FCFP6. The molecules must be pre-aligned when using Cresset's field/shape similarity.

Regression methods can be used when the biological activity data takes the form of real values (for example, pIC50, pKi) to calculate a function that can be used to predict the activity of new compounds. Classification methods are useful to analyze categorical data, i.e. when the biological data takes the form of qualitative labels (for example, 'high', 'medium', 'low), or of activity ranges (for example, ranges of % of inhibition) which can be appropriately converted into categories. Regression or classification models can be generated by setting the activity unit type in the node configuration dialog.

### Flare Score kNN

Calculates predicted activities for molecules against a kNN model. Except for 2D kNN models, the molecules must be pre-aligned in a manner consistent with the original model to make sensible predictions.

### Flare Field Descriptors

Generates Cresset 3D field descriptors for aligned ligands which can be used to build machine learning models.

The aligned set of molecules is used to derive a set of 'sample positions' around the molecules based on their field points, which can be used to probe each molecule for the electrostatic potential or for the volume taken up at those positions and optionally for the steric (van der Waals) and hydrophobic fields. The data matrix derived from the sample values can be exported into the KNIME workspace for use with machine learning data analysis methods.

### Flare Model Info

Extracts information on a Field QSAR or kNN model saved within a Flare project, including the settings used to create the model. The Flare project may have been created by the Flare Build node or read in by the Flare Project Reader node.

## ‘Project’ group

This group includes KNIME nodes for reading, writing, viewing and extracting information from Flare projects.

### Flare Project Reader

Reads a Flare project file which can be used as the input for Flare Score. The Flare project can be viewed using the Flare Project Viewer node.

### Flare Project Writer

Writes a Flare project generated with Flare Build to a file. The Flare application may then be used to view the Flare project outside of KNIME or the project can be used as an input to the Flare Score node.

### Flare Project Viewer

Launches Flare to view the input Flare project.

# Spark™

Spark finds biologically equivalent replacements for key moieties in your molecule and guides you through the process of growing and linking ligands. Given a 'starter molecule', preferably in the bioactive conformation, you can select a portion to replace and specify which databases to search. Spark will then present a list of biologically relevant replacements ranked using Cresset’s unique molecular field technology, or using Lead Finder’s docking score. As well as replacing central parts of a molecule, Spark can suggest replacements for terminal groups. It can be used to grow ligands and fragments into unoccupied pockets of the target protein, carry out ligand joining and macrocyclization experiment, and to find a fragment which displaces a crystallographic water molecule near your ligand. Unlike other fragment replacement methodologies, Spark works in product space so all potential replacements are merged into the original starter molecule before scoring, leading to more diverse yet realistic results.

Spark is supplied with several databases of moieties generated from fragmentation of commercially-available compounds, active molecules reported in the literature, available reagents, and theoretical studies. An optional database generator module enables the creation of custom fragment databases.

In KNIME the Spark nodes provide access to the scaffold hopping and R-group exploring capabilities with a specialized node to enable graphical specification of the portion of the starter molecule to be replaced. Lastly a database creation node is included.

### Spark Database Search

This node is the main Spark node and is used to search databases of fragments for biologically equivalent replacements (bioisosteres) for a portion of the starter molecule. If the starter molecule is supplied in 2D then it is first converted to the nearest 3D conformation before the search is performed.

Spark result molecules can be scored by ligand field similarity or by docking into a protein structure. Optionally, field and pharmacophore constraints, docking constraints and/or a protein structure (to be used as an excluded volume with ligand similarity scoring) can be specified to bias the Spark search and penalize results which do not satisfy the constraints. Additional reference molecules can be used to guide the experiment.

### Spark Fragment Selector

The Spark Fragment Selector enables the graphical selection of the portion to be replaced by Spark in a bioisostere replacement experiment. It takes the first molecule in the input and displays a dialog which allows the user to select the portion of the molecule to replace. The output of this node should be connected to the input of the Spark Database Search node.

### Generate Spark Database

Generate Spark Database is a tool for generating or updating Spark databases. It reads a list of molecules and breaks them into fragments and stores the fragments into a database file for use with the Spark Database Search node.

## XedTools™

The XedTools provide access to XED, one of the most innovative molecular mechanics force fields in existence. XED uses off atom charges on electronegative atoms to give a more accurate representation of the charge density surrounding an atom, resulting in an improved description of molecular interactions.

### XedMin

XedMin™ is a ligand minimizer that uses Cresset's proprietary XED force field. As well as free minimization a second input to the KNIME node enables a context for the optimization to be provided. Most commonly the context for the minimization will be the active site of a protein.

### XedeX

XedeX™ uses Cresset's XED force field to generate conformations for an input ligand. It uses a torsion randomisation and subsequent minimization algorithm to generate diverse, low energy conformations. In independent testing XedeX performs as well as the best available conformation hunting algorithms at reproducing bioactive conformations.

# Cresset Example Workflows

Example workflows that encompass each of the Cresset KNIME nodes are available to download at https://www.cresset-group.com/downloads/Cresset_KN_desktop_examples.zip.

Note that some Cresset example workflows may include optional third party KNIME nodes, which are only used to make the workflow run with minimal user intervention.

# Supported Platforms

Cresset KNIME nodes are supported under KNIME 4.0.0 and later releases. The tables below show the supported platforms for the KNIME nodes and the underlying command line binaries. Note that Spark requires access to Cresset supplied or internal Spark databases. The Cresset supplied databases require a minimum of 6GB of disk space up to a maximum of 40GB.

## Minimum specifications

| Hardware | Specifications |
|---|---|
| RAM | 4GB |
| Processor | Intel i7 processor or equivalent |
| Disk space | 6GB |

## Recommended specifications to install all binaries

| Hardware | Specifications |
|---|---|
| RAM | 16GB |
| Processor | Intel i7 processor or equivalent |
| Disk space | 16GB |

## Supported Operating Systems

| Windows® | Linux® |
|---|---|
| Windows 10<br>Windows 11 | Red Hat Enterprise Linux 8 |

Alternative operating systems for Flare and Spark include Red Hat Enterprise Linux 7.
**Note** that Red Hat Enterprise Linux 7 will no longer be supported after June 2024, as its maintenance support phase will come to an end.

# Installation

The Cresset KNIME nodes wrap functionality of key binaries and hence both the application binary and the nodes must be installed. Operation of the binaries is subject to a license agreement and requires a valid license file (see 'Licensing' section below).

## KNIME nodes

To install the Cresset KNIME nodes add the Cresset KNIME update site to your software sources. To do this, start KNIME, navigate to the Help menu, choose 'Install new software'. Now click 'Add' then in the Location box enter 'http://www.cresset-group.co.uk/knime/update'. Select the nodes that you wish to install and agree to the license agreement. Upon restarting KNIME, the nodes will appear under a top-level 'Cresset' directory.

If you are unable to access the Cresset hosted update site then we are happy to provide a zip file of the site for you to use internally. Please contact Cresset support for the current URL.

Detailed installation instructions for Cresset KNIME nodes can also be found at https://www.cresset-group.com/software/updating-cresset-knime/.

## Application binaries

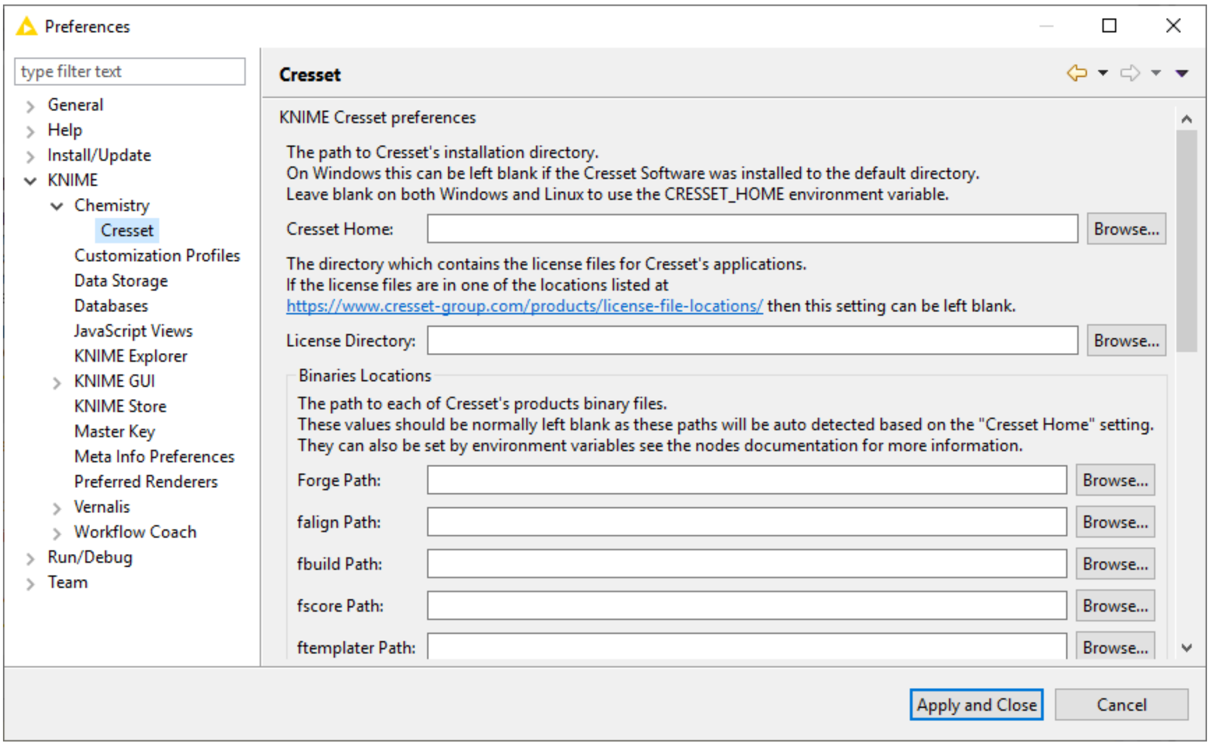
Installation of the Cresset application binaries is described in detail in the release notes for the individual binaries or in general at https://www.cresset-group.com/support/support-resources/installing-desktop-applications/.

## Cresset Engine Broker™

The Cresset Engine Broker can enable several nodes to use the power of your computing cluster to reduce the time it takes for the node to run. To use the Cresset Engine Broker set the CRESSET_BROKER environment variable to 'hostname:port' where 'hostname' is the name or IP address of the server running the Cresset Engine Broker and 'port' is the port the Cresset Engine Broker is running on.

## Environments

To execute correctly, each protocol requires the location of the corresponding binary to be set within KNIME. This can be configured in the KNIME preferences under the KNIME → Chemistry → Cresset menu.



Cresset

KNIME Cresset preferences

The path to Cresset's installation directory.
On Windows this can be left blank if the Cresset Software was installed to the default directory.
Leave blank on both Windows and Linux to use the CRESSET_HOME environment variable.

Cresset Home:

The directory which contains the license files for Cresset's applications.
If the license files are in one of the locations listed at https://www.cresset-group.com/products/license-file-locations/ then this setting can be left blank.

License Directory:

Binaries Locations

The path to each of Cresset's products binary files.
These values should be normally left blank as these paths will be auto detected based on the "Cresset Home" setting.
They can also be set by environment variables see the nodes documentation for more information.

Forge Path:

falign Path:

fbuild Path:

fscore Path:

ftemplater Path:

Alternatively, this can be achieved by setting the CRESSET_HOME environment variable to point to the installation directory for the Cresset binaries. Default values are shown in the table below.

| Operating System and Install level | Default Location |
|---|---|
| **Windows Global** | %PROGRAMFILES%\Cresset-BMD |
| **Windows Local install** | %APPDATA%\Cresset-BMD |
| **Linux global install from rpm** | /opt/cresset |
| **Linux local install from tgz or tar.xz** | $HOME/cresset |

In addition to setting CRESSET_HOME, the absolute path to each individual executable can be set using the variables shown below.

| Node | Environment variable |
|---|---|
| **Flare Viewer** | CRESSET_FLARE_EXE |
| **pyflare** | CRESSET_PYFLARE_EXE |
| **Spark Fragment Selector** | CRESSET_SPARK_EXE |
| **Spark Database Search** | CRESSET_SPARKCL_EXE |
| **Generate Spark Database** | CRESSET_SPARKDB_EXE |
| **XedeX** | CRESSET_XEDEX_EXE |
| **XedMin** | CRESSET_XEDMIN_EXE |
| **Cresset Engine Broker** | CRESSET_BROKER |

The licensing for the XedMin and XedeX binaries requires that either the 'cressetbmd_LICENSE' or 'XED_LICENSE' environment variable be set to the location of your license file.

Note also that the Spark command line binaries require access to database files or location in which to create databases. These are most commonly specified using the environment variables SPARK_CRESSET_DB and SPARK_DB respectively. Full details can be found in the application installation instructions.

## Licensing

The KNIME nodes require you to accept the terms of our license agreement when they are installed but do not require further license files. However, the Cresset software used by the KNIME nodes does require a valid license file or license server to operate correctly: if you are a customer and cannot locate your license file, then please contact Cresset support (support@cresset-group.com).

Alternatively, contact us at enquiries@cresset-group.com for more information. License files should be placed in one of the locations detailed at https://www.cresset-group.com/support/support-resources/license-file-locations/ or in the KNIME application directory.

# Upgrading Cresset Nodes for KNIME

## Upgrading from 2.7.0 to 2.8.0

- Cresset KNIME nodes 2.8.0 work optimally on KNIME 4.0.0 or newer
- Cresset KNIME Example workflows 2.8.0 require KNIME 4.5.2 or newer

## Upgrading from 2.6.0 to 2.7.0

- Cresset KNIME nodes 2.7.0 work optimally on KNIME 4.0.0 or newer
- Cresset KNIME Example workflows 2.7.0 require KNIME 4.5.2 or newer
- The Forge nodes listed below have been deprecated and replaced by Flare nodes with equivalent functionality:
    - Forge Build Field QSAR
    - Forge Score Field QSAR
    - Forge Build Activity Atlas
    - Forge Score Activity Atlas
    - Forge Model Info
    - Forge Project Viewer
    - Forge Project Reader
    - Forge Project Writer
    - Forge Align
- The Forge nodes listed below have been deprecated:
    - Forge Build Machine Learning
    - Forge Score Machine Learning
    - Forge Project Molecule Extractor
    - Forge Project Roles Extractor
    - Activity Miner
    - FieldTemplater
    - Surface Writer

## Upgrading from 2.5.0 to 2.6.0

- Cresset KNIME nodes 2.6.0 work optimally on KNIME 4.0.0 or newer.

## Upgrading from 2.4.0 to 2.5.0

- Cresset KNIME nodes 2.5.0 require KNIME 3.1.2 or newer.
- The ‘Forge Build k Nearest Neighbor’ has been deprecated, use the ‘Forge Build Machine Learning’ node instead.
- The ‘Forge Score k Nearest Neighbor’ has been deprecated, use the ‘Forge Score Machine Learning’ node instead.

## Upgrading from 2.3.0 to 2.4.0

- When opening old workflows the following error may appear in the console window.

  ```
  ERROR LoadWorkflowRunnable Status: Error: Unable to insert
  connection "[6(2) -> 8( 1)]"
  ```

  This happens as the type of connection for Forge projects was changed and the 'Torch/Forge Molecule Viewer' node cannot receive this new type of connection. To fix this problem, replace the 'Torch/Forge Molecule Viewer' node with the 'Forge Project Viewer' node.

- When opening old workflows the following warning may appear in the console window.

  ```
  WARN Align molecules using Forge Align Substituted
  'com.cressetgroup.knime.nodes.torch.TorchNodeFactory' for unknown factory
  'com.cressetgroup.torch.TorchNodeFactory'
  ```

  This warning can be safely ignored.

# Change Log

## V2.8.0

- Updated the Flare Docking node to include also the protein chains carrying cofactors ('Other' chains) and water molecules ('Water' chains) in the docking experiment, when available in the protein used for docking
- Updated the Flare Align node: when using a protein structure as an optional excluded volume, any protein chains carrying water molecules ('Water' chains) are now included in the excluded volume, together with any protein chains carrying cofactors (as in previous versions of this node)
- The Flare Viewer node now disconnects from the Flare session it has launched, enabling the user to reset the KNIME workflow and its individual nodes.

## V2.7.0

- New Flare nodes replacing legacy Forge nodes with equivalent functionality:
  - Flare Build Field QSAR: build Field QSAR models for activity
  - Flare Score Field QSAR: predict the activity of new compounds using a Field QSAR model pre-calculated with Flare or Flare Build Field QSAR
  - Flare Build Activity Atlas: summarize the SAR of a ligand series with Activity Atlas
  - Flare Score Activity Atlas: calculate the Novelty of new ligands using an Activity Atlas model pre-calculated with Flare or Flare Build Activity Atlas
  - Flare Model Info: Extract information on a Field QSAR or Machine Learning model saved within a Flare project
  - Flare Project Viewer: open Flare to view the input Flare project
  - Flare Project Reader: read a Flare project
  - Flare Project Writer: write a Flare project
  - Flare Align: align ligands in Flare to one or more reference compounds using Cresset ligand-based alignment
- New Flare Docking node to run docking and scoring experiments in Flare, including template docking and covalent docking
- New Flare Build kNN: build kNN regression and classification models for activity

- Flare Score kNN: predict the activity of new compounds using a kNN model pre-calculated with Flare or Flare Build kNN
- Flare Field Descriptors: generate Cresset 3D field descriptors for aligned ligands which can be used to build machine-learning models
- New Cresset example workflows to illustrate the functionality of Cresset KNIME nodes.

## V2.6.0

- New option to use docking instead of ligand similarity as the method for scoring the results molecules from a bioisostere replacement experiment in the Spark Database Search node.
- New option to set the docking region buffer size in the Spark Database Search node.
- New option to specify docking constraints in the Spark Database Search node.
- New option to set the maximum docking constraint penalty in the Spark Database Search node.
- Improved field and pharmacophore constraints, which can now be specified for both starter and reference molecules.
- Improved fragment selection, now supporting the selection of a disconnected fragment in the starter molecule to perform water replacement experiments.

## V2.5.0

- Dropped support for older versions of KNIME. KNIME version 3.1.2 or newer is now required.
- New Flare Viewer node to view ligands, proteins and results in Flare.
- New pyflare node to use the pyflare Python interpreter with ligands and proteins in a KNIME workflow.
- New Surface Writer node to write molecule surfaces to a directory.
- New Forge Build Machine Learning to create QSAR models using various Machine Learning methods.
- New Forge Score Machine Learning nodes to calculate predicted activities for molecules using a Machine Learning model.
- Deprecated the Forge Build k Nearest Neighbor (kNN) and Forge Score k Nearest Neighbor (kNN) nodes. The Forge Build Machine Learning and Forge Score Machine Learning nodes should be used instead.
- The new Weighted Sum Activity Cliffs Summary algorithm is now the default in the Forge Build Activity Atlas node.
- The XedMin, Forge Align, Flare Viewer and pyflare nodes now accept proteins in PDB format.
- New option to set pharmacophore constraints to the Forge Align node.
- New option to set field and pharmacophore constraints to the Spark Database Search node.
- New option to remove boats conformations before performing the alignment in the Forge Align node.
- In the Spark Database Search node, multiple reference molecules can now be used to help guide the search.
- In the Spark Database Search node, users can now specify a database to search in addition to the databases it autodetects.
- The Forge Project Roles Extractor node now also outputs additional data for each role.
- Forge Build Activity Atlas node now also writes the Activity Atlas surfaces to a directory.
- Added Weighting Scheme option to the Forge Build Machine Learning node in support of kNN models.

- The Forge Build Activity Atlas node can now read similarity data from a column in the input table.
- The Forge Align node and Spark Database Search node now support the Tanimoto and Tversky similarity metric in addition to the default Dice metric.
- The nodes Forge Align, Forge Build Machine Learning, Activity Miner, Forge Molecule Viewer and Generate Spark Database now accept molecules in SMILES format.
- The 'Torch/Forge Molecule Viewer' node has been renamed to 'Forge Molecule Viewer'.
- Fixed Field Engines not being stopped when a node is cancelled.
- Fixed bug which prevented 2D kNN models from being built.

## V2.4.0

- New preferences page under KNIME → Chemistry → Cresset for configuring binary locations and FieldEngine settings.
- Added new nodes for Activity Atlas and k Nearest Neighbors.
- Added new 'Forge Project Roles Extractor' and 'Forge Project Molecules Extractor' nodes for extracting data out of a Forge project.
- Added setting for "Gradient Cut-off" to the 'Spark Database Search' node.
- The nodes ports for Forge projects now use a special kind of connection.
- Drag-and-drop of Forge and Torch project files onto a workflow automatically creates a 'Forge Project Reader' node.

## V2.3.0

- New Forge model info node that reports the statistics and log file associated with a Forge model.
- Added support for Cresset's engine broker environment for distribution of calculations to a Linux cluster.
- New options added to the Forge Align and FieldTemplater nodes to control conformation hunting.
- New option added to Spark database generator node to process molecules using all the available reagent rules.

## V2.2.0

- New Generate Spark Database node which can generate Spark Databases to be used with the Spark Database Search node.
- New FieldTemplater node which can compare molecules using their electrostatic and hydrophobic fields in order to find common patterns.
- Changed the output of the Activity Miner node so that the resulting columns are appended to the input columns instead of replacing the input columns.
- The Activity Miner node now has an additional output containing a Forge project. The project can be viewed using the Activity Miner GUI by using the Torch/Forge Viewer node with the 'Open Activity Miner' checkbox ticked. The project can also be written to a file using the Forge Project Writer node.
- Added Read Settings to the Torch/Forge Viewer node which define how molecules should be read in e.g. as conformations or aligned molecules.

- Added ‘Only score molecules’ option to Forge Align node. If selected then conformations will not be generated and the molecules will not be moved.
- Added ‘Skip Conformation Hunt’ option to Forge Align node.
- Added the ‘Attachment types’ column to the Spark Database Search node output.
- Added Cresset Engine Broker Support to Forge Align, Forge Build, Activity Miner and FieldTemplater nodes.
- Forge Align now defaults to 50 conformations instead of 100 when ‘quick’ mode is selected.
- Spark Database Search can now list more than 10 databases.
- Fixed an error being displayed when changing the speed option in Forge Align.
- Fixed the ‘Conformers of achiral molecules can be inverted’ in Forge Align doing the opposite of what the help text describes.
- Fixed the ‘Add enantiomeric conformations’ in XedeX doing the opposite of what the help text describes.
- Fixed the ‘ring library’ in XedeX doing the opposite of what the help text describes.
- Fixed the Fragment SMILES output column appearing blank when a KNIME plugin present which provides a SMILES render.

## V2.1.0

- New Activity Miner node (part of the Forge collection) to calculate similarity and disparity matrices.
- Forge_align node now correctly outputs additional alignments when requested.
- Spark database search node can now be incorporated into a workflow in the absence of installed databases.
- Forge_build node includes new option to specify activity as log(Ki).
- Added option to control the number of processors (FieldEngines) used by Spark search node.

## V2.0.0

- New nodes to wrap all Forge functionality and Spark search functionality.
- Previous TorchLite node replaced by generic Torch/Forge viewer node that uses either Torch, TorchLite or Forge as the molecule viewer.
- XedMin node now capable of minimising ligands within a protein cavity.
- Introduced update site to facilitate software updates and smooth corporate installations.

## V1.1.0

- Updated nodes to wrap new Forge Align and TorchLite binaries.
- Introduced CRESSET_FORGEALIGN_EXE and CRESSET_TORCH_EXE variables to enable use of custom install locations and binaries.
- Introduced automatic recognition of CRESSET_HOME for standard installations.

# Known Issues and Bugs

During the installation of the Cresset KNIME Nodes, KNIME may show a message informing that errors occurred while reading the file signatures. This message is harmless and can be ignored: Cresset nodes and workflows will still work correctly in KNIME.

# Support

Cresset
New Cambridge House
Bassingbourn Road
Litlington
Cambridgeshire, SG8 0SS, UK

+44 (0)1223 858890
support@cresset-group.com
www.cresset-group.com
Twitter: @cressetgroup
LinkedIn: Cresset group