

A close-up photograph of a person's hand holding a complex, multi-colored protein structure model. The model features various helices and loops in shades of purple, blue, green, orange, and red, with a cluster of grey spheres at the top. The background is a blurred image of a person's torso and arm.

SCHRÖDINGER®

KNIME workflows in LiveDesign FAQ

Jean-Christophe Mozziconacci, KNIME extension Product Manager

- LiveDesign node usage
- Diagnose KNIME LiveDesign model issues
- Create a KNIME LiveDesign model
 - Choose the generic protocol
- Use third-party tools and extensions in KNIME models
- Run models on the KNIME server
- Control workflow parameters or change system specific files
- Miscellaneous

KNIME workflow in LiveDesign – documentations and materials

- Documentation

https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Support_materials/

- General presentation
- FAQ
- Cheat sheets
- Template and tutorial workflows

- Validated models

- Overview and download

https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/

- Description and screenshots

https://api.hub.knime.com/repository//Users/schroedinger/LiveDesign_models/Validated_KNIME_models.pdf:data

- New feature presentation

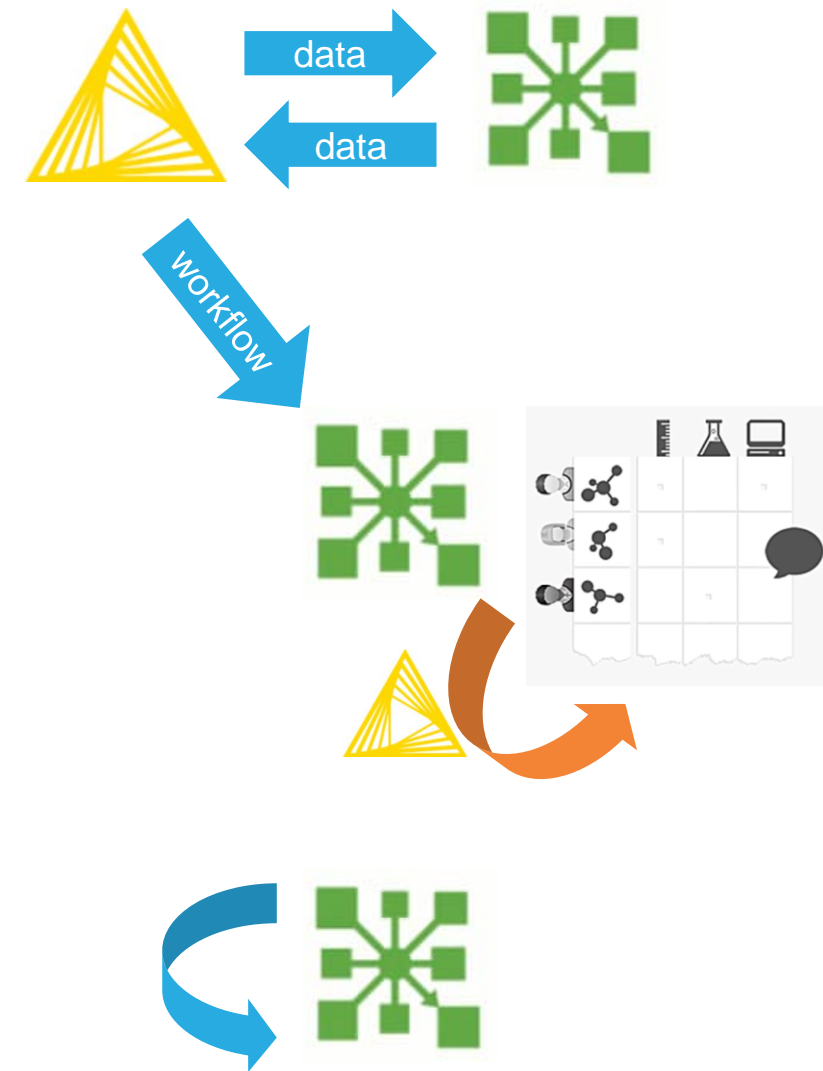
https://www.schrodinger.com/knimeworkflows/KNIME_newfeatures.pdf

- KNIME workflow examples

https://hub.knime.com/schroedinger/spaces/Workflow_examples/latest/

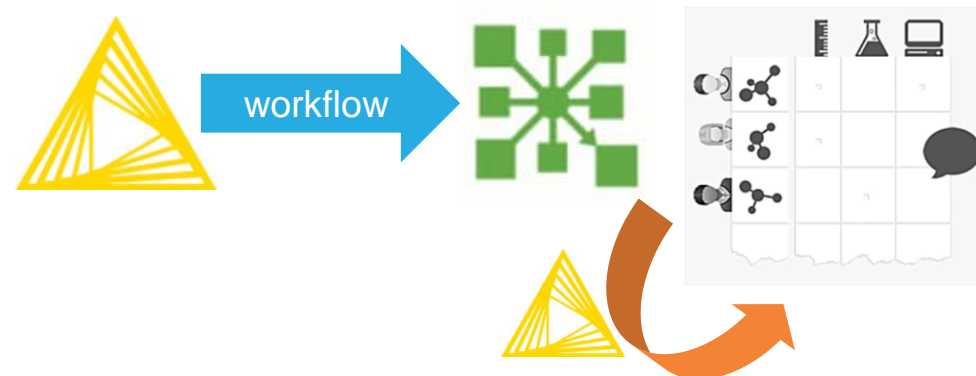
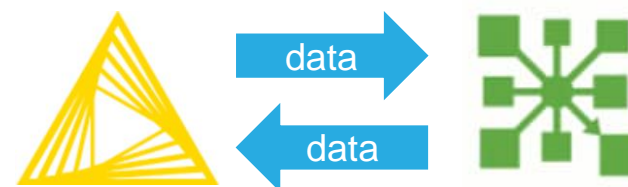
LiveDesign Nodes in KNIME

- **Import from LiveDesign**
 - Populates a KNIME table with data from a LiveReport
- **Export to LiveDesign**
 - Sends data from a KNIME table to a LiveReport
- **Export images to LiveDesign**
 - PNG or SVG images stored in Free Form Columns
- **Upload a workflow as LiveDesign model**
 - Creates a model which executes the KNIME workflow in LiveDesign
(no need to open the LiveDesign administration panel)
 - The model is now ready to be run on any LiveReport and the output will be added as new column(s)
- **Administrate LiveDesign models and protocols**
 - eg backup copies, testing copies, copy from testing to production instance, new version deployment, version comparison



LiveDesign Nodes in KNIME – usage

- **Import > Export to LiveDesign**
 - Calculations on your local KNIME installation
 - Possibly interactive data manipulation, workflow adapted to the data
 - Import from LiveDesign and Export to LiveDesign nodes
- **KNIME models**
 - Calculations on the LiveDesign machine
 - Available to other LiveDesign users
 - Completely automated, starting from 1 column and creating some in the same LiveReport
 - Upload as LiveDesign model





How to diagnose issues with the Upload
as LiveDesign model node and KNIME
model failures?

Diagnose failures – Upload as a LiveDesign model node

Error messages in the console and/or at the bottom of the panel:

- 401 Client Error: Unauthorized for url:
<https://<myHost>.onschrodinger.com/livedesign/api/auth/login>
 - Wrong credentials
- Username or password is not set.. Could not get LiveDesign credentials.
 - The password is not set in the upstream Credential configuration node
- Got a null credentials provider.. Could not get LiveDesign credentials
 - You need to connect the node to an upstream node (we're going to change this misleading error message).
- LiveDesign version SNAPSHOT is not supported
 - You've connected to a different LiveDesign version in the current KNIME session. You need to close and restart KNIME to be able to connect (same as in Maestro).

Diagnose failures – Upload as a LiveDesign model node

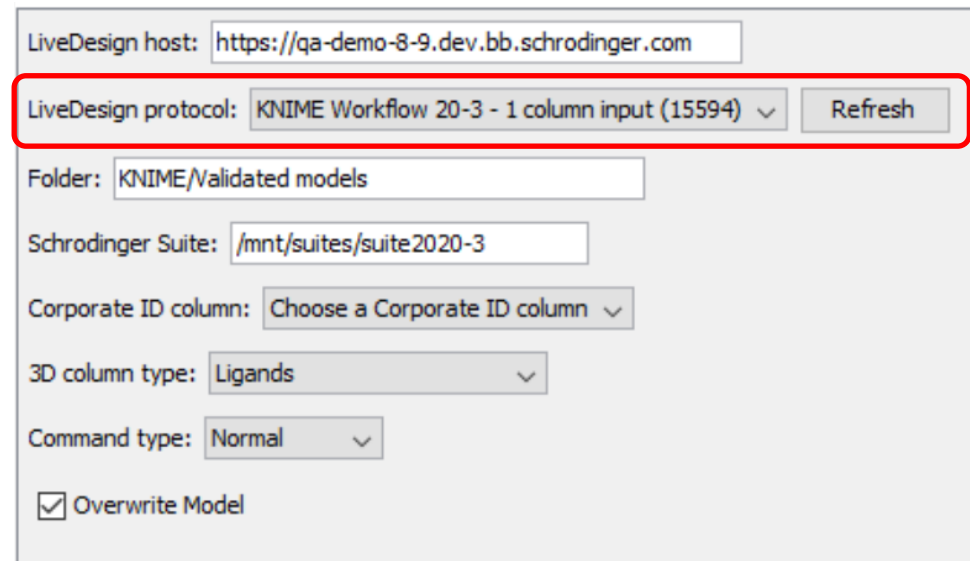
Error messages in the console and/or at the bottom of the panel:

- HTTPSConnectionPool... Max retries exceeded with url... (Caused by ConnectTimeoutError: The LiveDesign instance can't be reached
 - Possibly a network or VPN issue
- Unable to extract the necessary files from the fetched tar file
 - Check the LiveDesign service isn't down while the machine is up and running (when you see this '502 Service is restarting' while connecting to the instance)

How to create a KNIME LiveDesign model?

Upload as LiveDesign mode node – Protocol choice

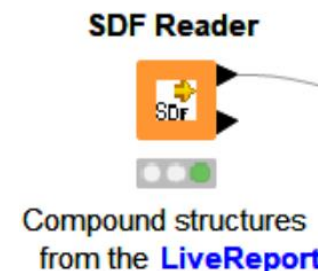
- Between the 2 Generic protocols
 - KNIME Workflow 20-4**
 - Takes the 'Compound Structure' LiveReport column (ligands in 2D) as input
 - KNIME Workflow 20-4 - 1 column input**
 - Takes 1 column from the LiveReport as input
eg a string, 3D structure, FEP map column
 - The input column is selected in the Parametrized model panel when executing the model
- Or use these generic protocols as template to create a customized protocol for specific scenarios



The screenshot shows the configuration panel for a LiveDesign node. A red rectangular box highlights the 'LiveDesign protocol' dropdown menu, which is currently set to 'KNIME Workflow 20-3 - 1 column input (15594)'. To the right of the dropdown is a 'Refresh' button. Other visible fields include 'LiveDesign host' (https://qa-demo-8-9.dev.bb.schrodinger.com), 'Folder' (KNIME/Validated models), 'Schrodinger Suite' (/mnt/suites/suite2020-3), 'Corporate ID column' (Choose a Corporate ID column), '3D column type' (Ligands), 'Command type' (Normal), and a checked 'Overwrite Model' checkbox.

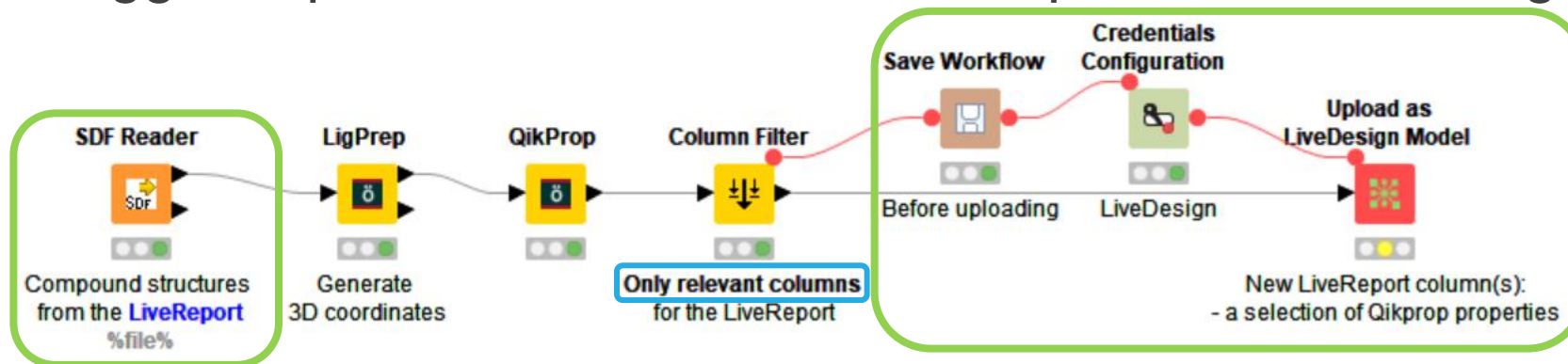
KNIME LiveDesign model – creation

- To use **KNIME Workflow** generic protocol
 - Create and test your workflow in KNIME
 - Use the KNIME **SDF reader** as input <https://kni.me/n/ZVPNMGw17wO0l1jZ>
 - It can be configured to read any file on disk
or read the default LiveDesign file included in the workflow examples:
`knime://knime.workflow/_Files/input_from_LD.sdf`
 - Add the “**LiveReport**” tag to the node comment
so the file initially read is replaced with the LiveReport structures when the workflow is executed in LiveDesign
 - Filter the table columns to keep only Corporate ID one and those to add to the LiveReport
 - Add the **Upload as LiveDesign model node** at the end of the workflow
 - Select KNIME Workflow [2X-X] as protocol
- To use **KNIME Workflow - 1 column input** generic protocol
 - Use one of the following templates to create your workflow:
 - https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/_Support_materials/Templates/
 - If you need specific data to configure your workflow:
 - Add the Breakpoint node to the template model and run it on your LiveReport
 - Use as template the executed workflow temporary file from LiveDesign admin Tasks
- Use these generic protocols as template to create customized protocols for specific scenarios



KNIME LiveDesign model – creation

- Tagged input SDF reader node and Upload as LiveDesign model node as output



Read molecules - 0:2...

Row ID	SDF Molecule	Molecule name
Row0	Molecule: V64...	V64713
Row1	Molecule: V64...	V64719
Row2	Molecule: V64...	V64721

Filtered table - 0:38 - Column Filter (Only relev...

Row ID	CT	Molecule name	QikPropBB
Row0	Molecule: V64713 #atoms:...	V64713	-0.694
Row1	Molecule: V64719 #atoms:...	V64719	-1.377
Row2	Molecule: V64721 #atoms:...	V64721	-0.201

Dialog - 2:181 - Credentials Input (Live Design)

Label: LD credentials

Description: Enter Description

Variable Name: ld_credentials

Parameter Name: ld_credentials

Username: Jean-christophe.mozziconacci

Password:

☒ Prompt user name in wrapped metanode dialog/wizard

☐ Save password in configuration (weakly encrypted)

☐ Use KNIME Server Login (when run on server)

Dialog - 0:140 - Upload as LiveDesign Mod...

LiveDesign host: https://qa-demo-8-9.dev.bb.schrodinger.com

LiveDesign protocol: Choose a protocol Refresh

Folder: KNIME

Schrodinger Suite: /mnt/suites/suite2020-3-099

Corporate ID column: Molecule name

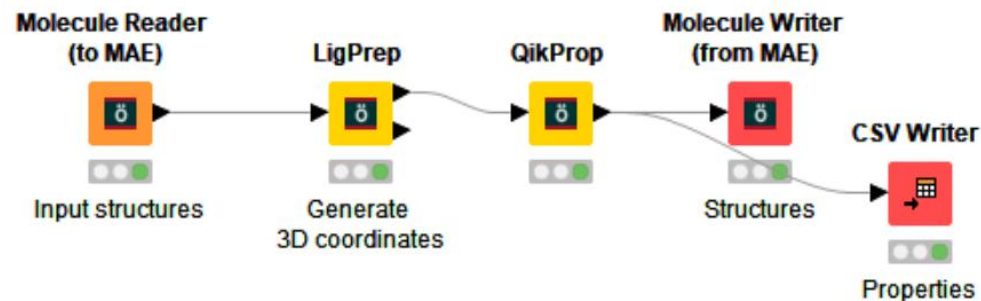
3D column type: Ligands

Command type: Normal

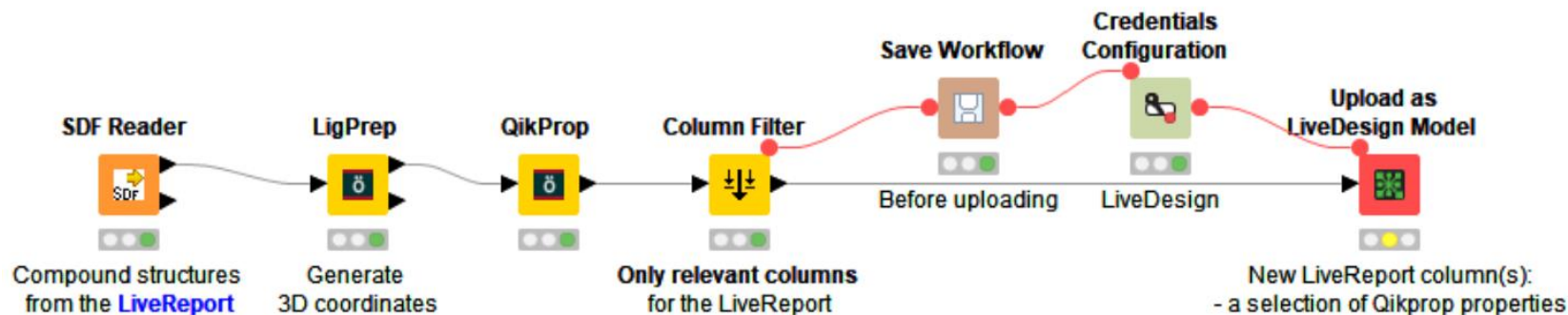
☒ Overwrite Model

KNIME LiveDesign model – creation tutorial

- To be prepared to be run in LiveDesign



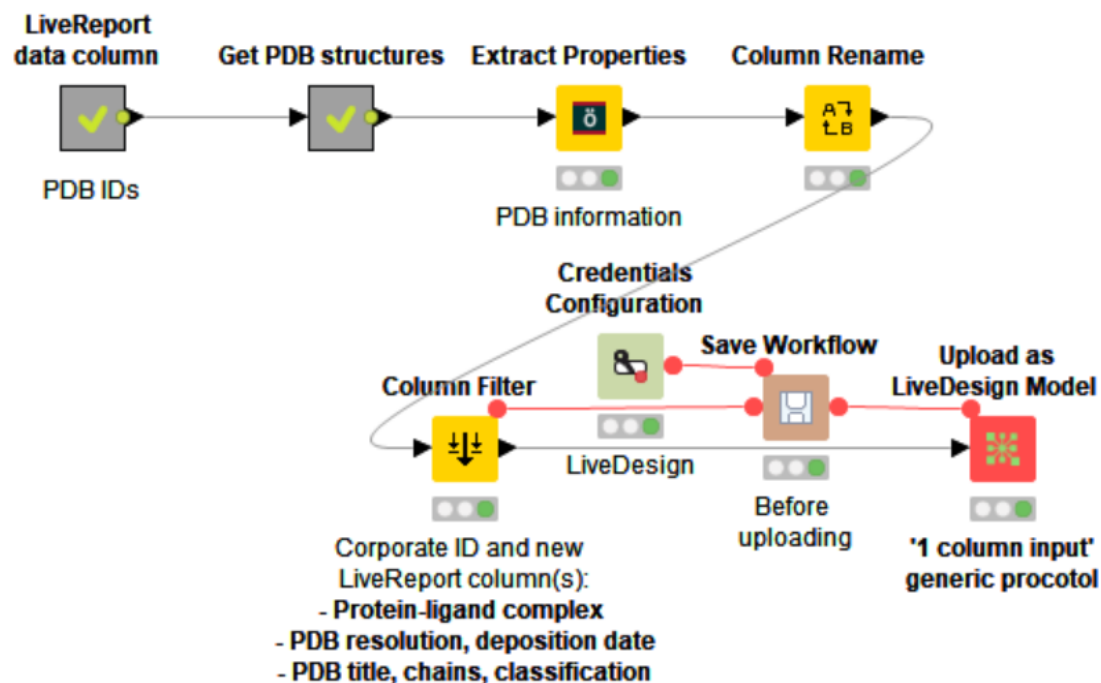
- Ready to be deployed to LiveDesign



Find workflows under https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Support_materials/Tutorials/

KNIME LiveDesign model – Parameterized model examples

- Get PDB structures
 - From a PDB IDs
 - Download the PDB structures
 - Extract structural information



Parameterize Model

Name:

Description:

Model Data

Input column: Change Column...

☐ Publish(share data between Live Reports)(?)

In this Live Report

Other Columns

Solubility QSAR (ph7.4) (LogS)

Membrane (dG_insert)

ADME profile

Thrombin Docking (Ligand Interaction Diagram)

Thrombin Docking (3D)

Thrombin Docking (Run Status)

Thrombin WaterMap (3D)

Thrombin Builder (3D)

ideas

Run FEP

Modeling Status

Recommend for Synthesis

Restricted Column (undefined) 17146

PDB ID (PDB ID)

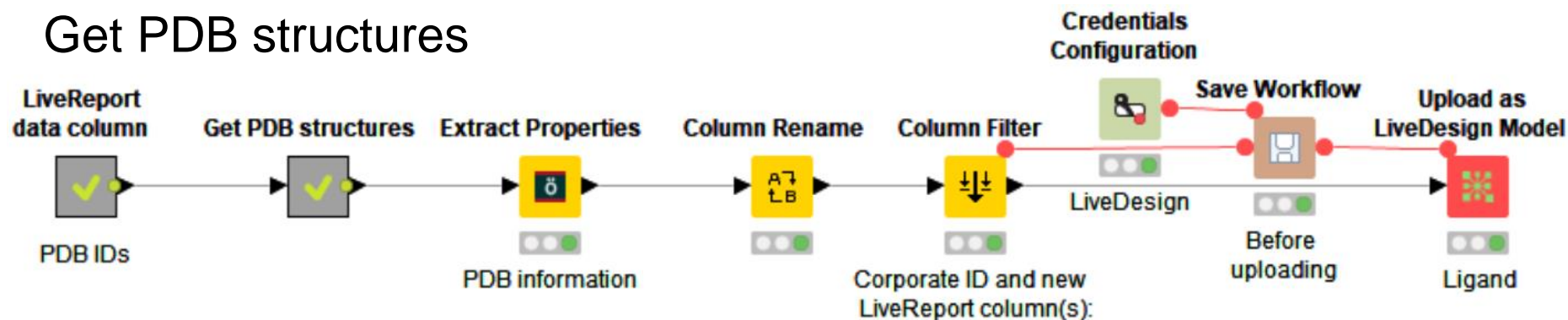
<

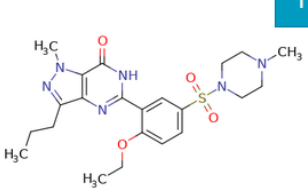

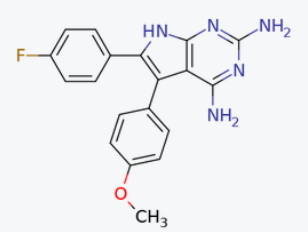

>

OK

KNIME LiveDesign model – Parameterized model examples

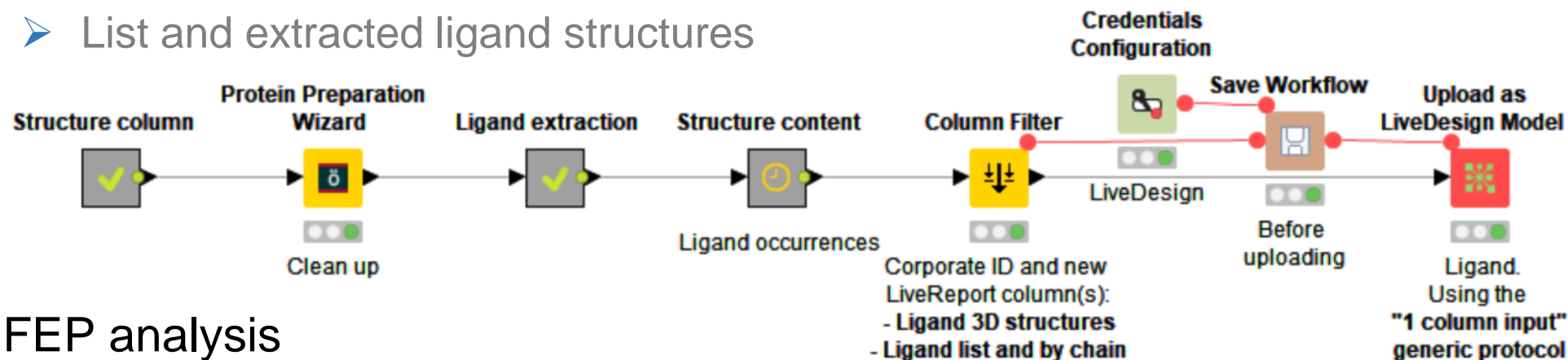
- Get PDB structures



Open Live Report + PDB										
	Compound Structure	1 ID	PDB ID (PDB ID)	FFC	Get PDB (PDB classification)	Get PDB (3D)	Get PDB (Deposition date)	Get PDB (PDB title)	Get PDB (Chain list)	Get PDB (Resolution)
1		V38532	1TBF 1TBF	1ETR	HYDROLASE		20-MAY-04	CATALYTIC DOMAIN OF HUMAN PHOSPHODIESTERASE 5A IN COMPLEX WITH SILDENAFIL	A	1.3
2		V222790	4CMG	4CMG	OXIDOREDUCT		16-JAN-14	CRYSTAL STRUCTURE OF PTERIDINE REDUCTASE 1 (PTR1) FROM TRYPANOSOMA BRUCEI IN TERNARY COMPLEX WITH COFACTOR AND INHIBITOR	A, B, C, D	2.0

KNIME LiveDesign model – Parameterized model examples

- Cocrystallized ligands
 - From a PDB structure column
 - Protein preparation
 - List and extracted ligand structures



- FEP analysis
 - FEP map reader node
 - FEP map convergence information

The screenshot shows the 'COX FEP' Live Report interface. It includes a sidebar with navigation icons, a top bar with 'COX FEP' and 'Give Feedback' buttons, and a main table displaying FEP results for two compounds, V217803 and V217804. The table columns include Compound Structure, ID, fmp FEP map (.fmp), FEP1 (Energy Conv.), FEP1 (CCC Conv.), FEP1 (Lig. RMSD), and FEP1 (REST Exch.).

	Compound Structure	ID	fmp FEP map (.fmp)	FEP1 (Energy Conv.)	FEP1 (CCC Conv.)	FEP1 (Lig. RMSD)	FEP1 (REST Exch.)
6		V217803		Fair	Good	Good	Fair
7		V217804		Fair	Good	Good	Fair

How to use third-party tools and extensions in KNIME models?

Run third-party tools in KNIME

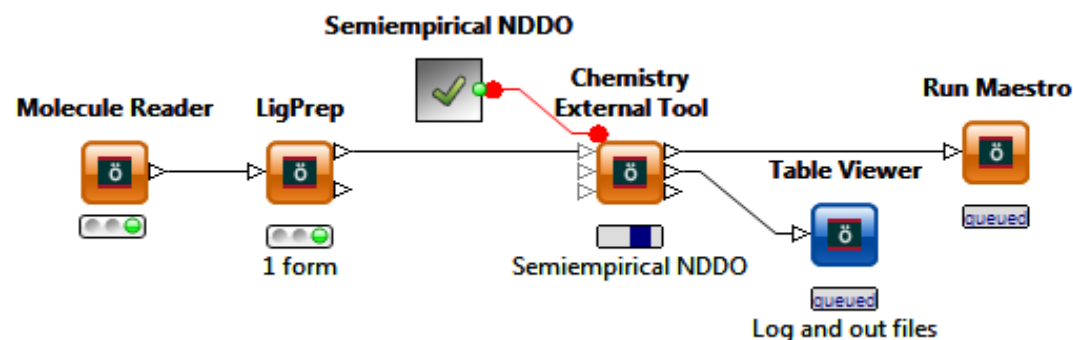
- **Community and Partner nodes**
 - Community Extensions - <https://update.knime.com/community-contributions/trusted/4.2> (to be opened in KNIME)
 - RDKit, Vernalis, Erlwood (Lilly), CDK, Genentech, R scripting...
 - Partner Extensions - <https://update.knime.com/partner/4.2>
 - ChemAxon/Infocom, LigandScout, MOE, Pharmacelera, Spotfire...
- **Command line tools can be run in the Chemistry external tool node**
 - <https://hub.knime.com/schroedinger/extensions/com.schrodinger.knime.feature.nodes/latest/com.schrodinger.knime.node.chemexttool.ChemExternalToolNodeFactory>
- **Python scripts can be run in the Python nodes or in the Chemistry external tool node**
 - Schrodinger Python nodes use the Python installation embedded in the Schrodinger Suite
<https://hub.knime.com/schroedinger/extensions/com.schrodinger.knime.feature.nodes/latest/com.schrodinger.knime.node.python11.Python11NodeFactory>
 - You can also use your own Python libraries
<https://www.schrodinger.com/kb/1072>
 - Workflow examples:
https://hub.knime.com/schroedinger/spaces/Workflow_examples/latest/Scripting/

How to run other KNIME extension nodes and licensed tools

- Install them in KNIME in the Schrodinger installation on the LiveDesign machine
 - The extensions can be installed using `$SCHRODINGER/KNIME_install.py`
- Install them in a specific folder on the LiveDesign machine
 - Possibly just before running the model
 - Use `-configuration ~taskd/prod_scripts/<my KNIME model>/config`
- Use a stand-alone KNIME installation on the LiveDesign machine
 - The installation folder can be copied from location machine
 - `KNIME_batch.py -knimeinstalldir` to point to the stand-alone installation
- Run the workflow on a KNIME server
 - Central extension installation location
 - Use the computational power and license server on the KNIME server
- Run the KNIME workflow on a remote cluster
 - Instead of running on the LiveDesign machine (computational power, licenses)
- Chemistry external tool node running remotely
 - **Under development**

New Chemistry external tool node

- **Optional input/output ports**, output column structure options, column name.
- Reads maegz files, input/output pdb, output Surface type
- **Flow variables, accessible by name**
- **Basename keyword, add extra columns** to the output



The screenshot shows the 'Job Manager Selection' dialog box. It has two tabs: 'Quickforms' and 'Job Manager Selection'. The 'Quickforms' tab is active. It contains the following fields:

- Method:** A radio button selection with options: RM1 (selected), PM3, AM1, MNDO, and MNDOd.
- Optimize geometry:** A checkbox that is checked.
- Keywords:** A text field containing the string 'esp mullik bonds pi super plotesp'.
- Host:** A text field containing the string 'localhost'.

The screenshot shows the 'ChemExternalTool' configuration window. It has three tabs: 'ChemExternalTool', 'Flow Variables', and 'Memory Policy'. The 'ChemExternalTool' tab is active. It contains the following fields:

- Input columns:** Three input fields labeled 'Input1 (%input_1%):', 'Input2 (%input_2%):', and 'Input3 (%input_3%):'. Each field has a dropdown menu. The first dropdown is set to 'CT'.
- Run command line for each input row:** A checkbox that is unchecked.
- Output replaces Input:** A dropdown menu set to 'Maestro'.
- Output only:** A dropdown menu set to 'Text'.
- log:** A checkbox that is checked.
- Command line(s):** A text area containing the following command:

```
$SCHRODINGER/run semi_emp.py -WAIT -HOST %HOST% %Options% -jobname %basename_1% %input_1%  
mv %basename_1%_out.mae %output_1%  
cat %basename_1%.log %basename_1%_1.out %basename_1%_2.out > %output_2%
```

Chemistry external tool nodes

Dialog - 5:42 - Chemistry External Tool 1:1 (Parse the log file)

File

ChemExternalTool11 | Flow Variables | General Node Settings

Output Column Structure

☒ Input plus Output
☐ Output replaces Input
☐ Output only

Column containing input (%input_1%): Log Log

☒ Run command line for each input row

Output Type (%output_1%): Text

☒ Output Column Name: Text

Command line(s):

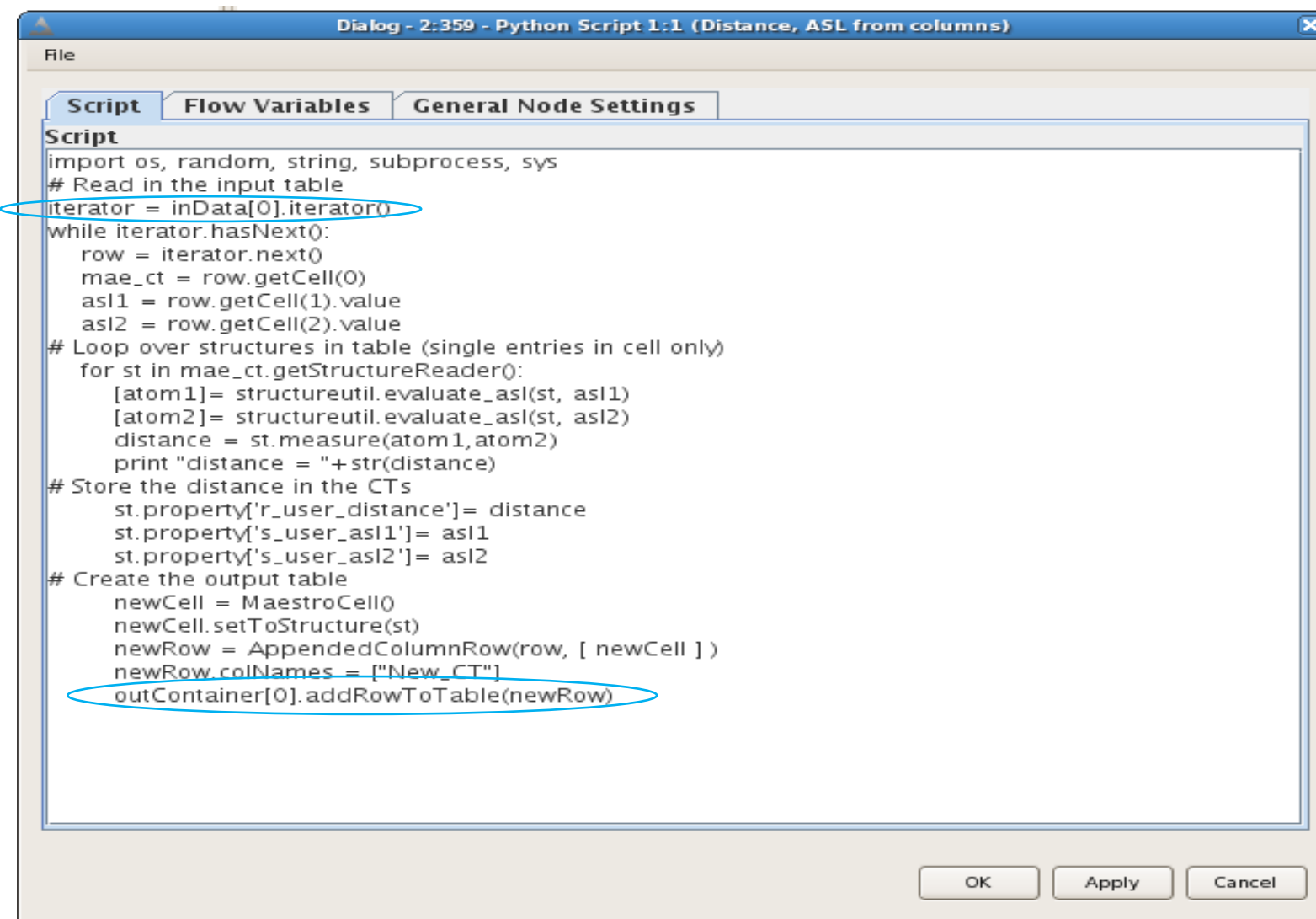
```
grep "high, medium" %input_1% > %output_1%  
sort %output_1% > tmp.txt  
uniq tmp.txt > %output_1%  
rm tmp.txt
```

(Use the string '%input_1%' for specifying the input file,
the string '%flow_<n>%', where <n> is 1, 2, or 3 for specifying the flow variables flow_<n>,
the string '%row_id%' for specifying the row id (only for each row),
and '%output_1%' for specifying the output file)

OK Apply Cancel

- Input/output types:
 - Maestro, SD, mol2, Smiles
 - Double, Integer
 - String, Text
 - Sequence, alignment
 - FingerPrint, Canvas Matrix
 - Phase Hypothesis, Glide Grid

Python nodes



```
Dialog - 2:359 - Python Script 1:1 (Distance, ASL from columns)

File

Script Flow Variables General Node Settings

Script

import os, random, string, subprocess, sys
# Read in the input table
iterator = inData[0].iterator()
while iterator.hasNext():
    row = iterator.next()
    mae_ct = row.getCell(0)
    asl1 = row.getCell(1).value
    asl2 = row.getCell(2).value
# Loop over structures in table (single entries in cell only)
for st in mae_ct.getStructureReader():
    [atom1]= structureutil.evaluate_asl(st, asl1)
    [atom2]= structureutil.evaluate_asl(st, asl2)
    distance = st.measure(atom1,atom2)
    print "distance = "+str(distance)
# Store the distance in the CTs
st.property['r_user_distance']= distance
st.property['s_user_asl1']= asl1
st.property['s_user_asl2']= asl2
# Create the output table
newCell = MaestroCell()
newCell.setToStructure(st)
newRow = AppendedColumnRow(row, [ newCell ])
newRow.colNames = ["New_CT"]
outContainer[0].addRowToTable(newRow)

OK Apply Cancel
```

- Input/output connectors: 0:1, 1:1, 1:2, 2:2
- Schrödinger's libraries
- Possibility to include third party libraries

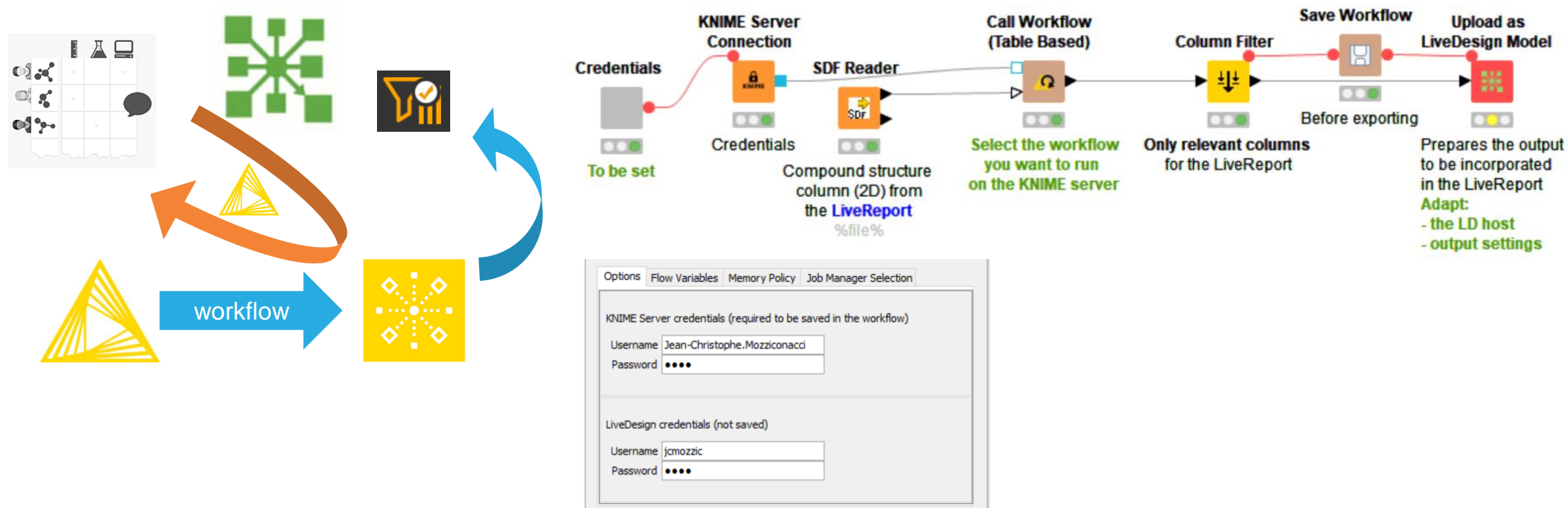


Run models on the KNIME server

Run a workflow on a KNIME Server from LiveDesign

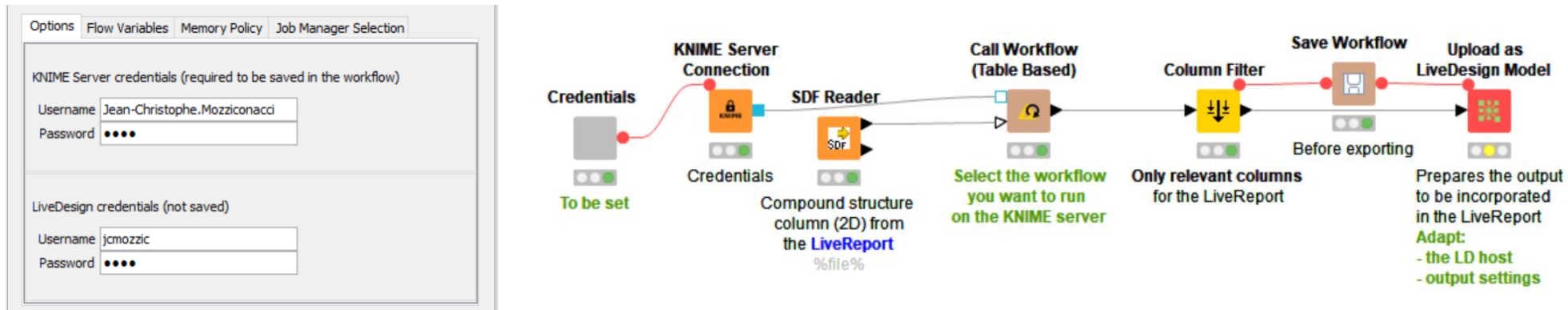
- Using the Upload as LiveDesign model node:
 - Configured to use the generic KNIME Workflow protocol
 - See example the follow validated model running a workflow on a KNIME server:

https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Run%20on%20a%20KNIME%20Server/My%20workflow%200running%20on%20a%20KNIME%20server



Run a workflow on a KNIME Server from LiveDesign – configuration

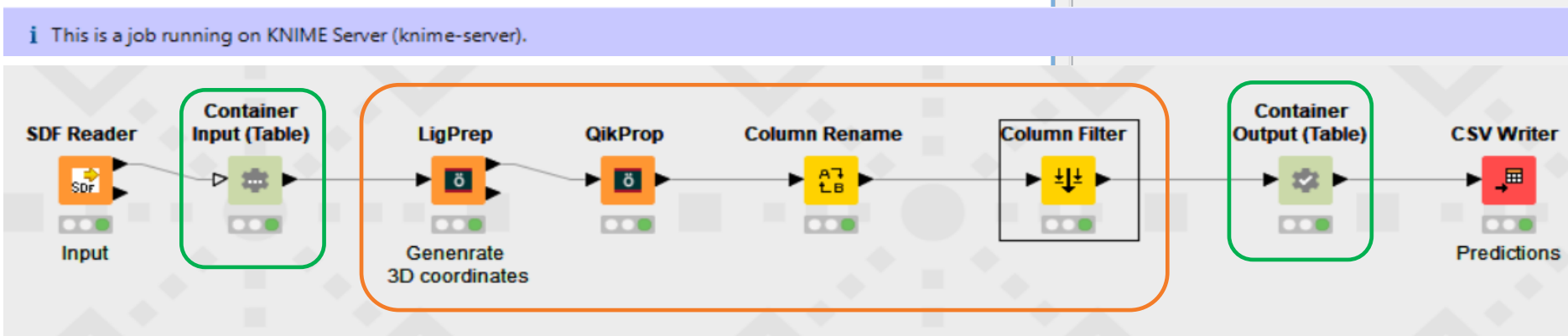
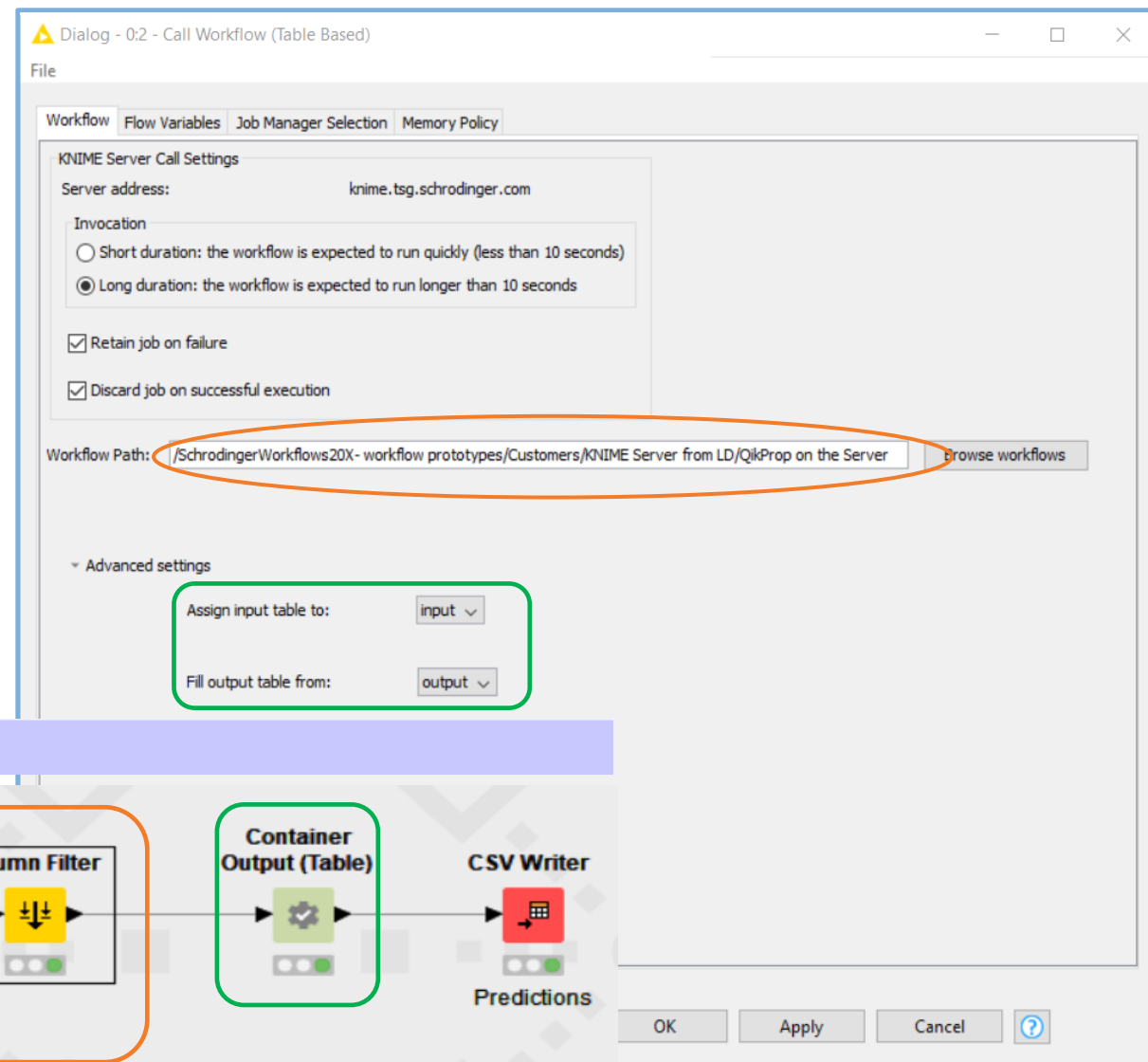
- Set the credentials
- Set the KNIME server address in the KNIME server connection node



- Select the location of the workflow to be executed on your KNIME server
 - Executing the Call workflow node will run the workflow on the KNIME server
- Configure the Upload node
 - Run the LiveDesign model created by this node
 - Failure reported in the Task log may reveal connection issue between the LiveDesign instance and the KNIME server

KNIME server workflow

- Call workflow (table based) node configuration:
 - Select the location of the workflow to be executed on your KNIME server
- Example of workflow to be run on the KNIME server
 - It must include the Container Input and output nodes to communicate with the Call workflow node



KNIME server from LiveDesign – benefits

- Workflows
 - stored, shared and modified on the KNIME Server
- Extra extensions
 - Installed and maintained from the KNIME Server
 - Otherwise Schrodinger installation change required at each release
- Access to corporate data, databases, tools and license servers
 - Set up and maintained from the KNIME Server
- Computational expensive calculations
 - Limited resources on the LiveDesign instance
 - No need to setup access to external computational resources
- More suitable to be called from the Gadgets



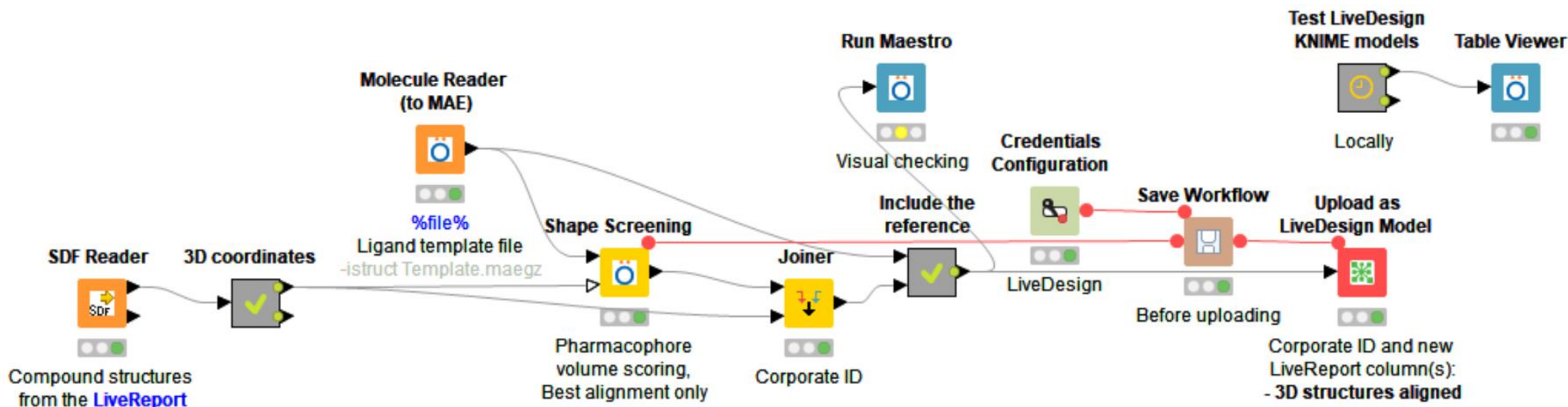
Control workflow parameters or change
system specific files

Customize a KNIME LiveDesign model – change files

- System specific files can be changed from LiveDesign admin page
 - No need to open KNIME, nor set it each time in the Parameterize model configuration panel
 - eg ligand alignment template, Glide grid, QSAR model file

MODEL DATA			
NAME	PARAMETER TYPE	DATA	PARAMETER SOURCE
extra file 1	File	Parent File: <i>mockup.txt</i> Currently: <i>Template.maegz</i> <input type="checkbox"/> Clear Change: <input type="button" value="Browse..."/> No file selected.	<input type="text" value="Set Fixed"/>
extra argument 1	Text	<input type="text" value="-istruct Template.maegz"/>	<input type="text" value="Set Fixed"/>

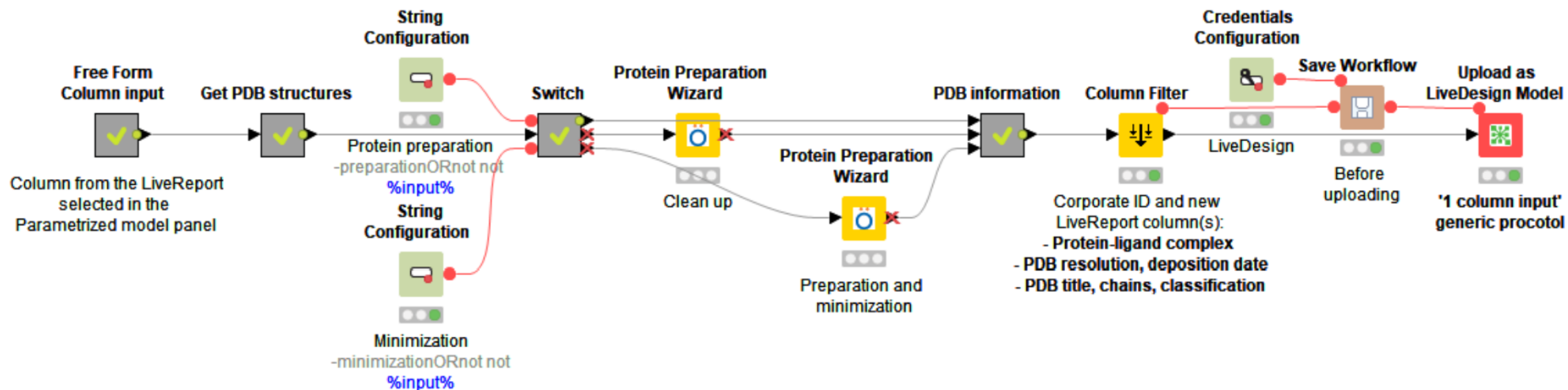
eg Alignment on a cocrystallized ligand:



Customize a KNIME LiveDesign model – control parameters

- Workflow parameters
 - Using the extra argument fields
eg Get PDB and preparation

MODEL DATA		
NAME	PARAMETER TYPE	DATA
extra argument 1 (48323)		
extra argument 1	Text	-preparationORnot preparati



Customize a KNIME LiveDesign model

COX

DATA & COLUMNS

Project LiveReport

PPrep

Computational Models

KNIME

Validated models

Get PDB from FFC and PPrep

3D

Chain list

Deposition date

PDB classification

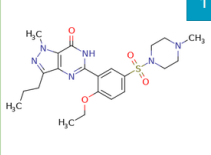

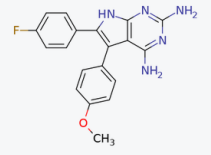

PDB title

prepared_with_version

Resolution

Open Live Report

PDB

Compound Structure	ID	A mini (Chain list)	A mini (Resolution)	A mini (PDB classification)	A mini (3D)	A mini (Deposition date)	A mini (prepared_with_version)	A mini (PDB title)
<div><div>1</div><div></div></div>	V38532	L, H	2.2	HYDROLASE/H INHIBIT		06-JUL-92	2020-4	REFINED 2 ANGSTROM X-RAY CRYSTAL STRUCTURE BOVINE THROMBIN COMPLEX FORMED WITH BENZAMIDINE More available...
<div><div>2</div><div></div></div>	V222790	L, H	2.5	HYDROLASE/H INHIBIT		06-JUL-92	2020-4	REFINED 2 ANGSTROM X-RAY CRYSTAL STRUCTURE BOVINE THROMBIN COMPLEX FORMED WITH BENZAMIDINE More available...

LiveDesign Admin

WELCOME, DEMO, STATUS / HELP / LOG OUT

Home > Models > Get PDB from FFC and PPrep (18271)

Change Model

ARCHIVE HISTORY

ID: 18271

Name: Get PDB from FFC and PPrep

Give this a short and descriptive name that modelers will understand

MODEL DATA	NAME	PARAMETER TYPE	DATA	PARAMETER SOURCE
extra argument 1 (48323)				
extra argument 1		Text	-preparationORnot preparat	Set Fixed
Input column (48324)				

Give Feedback

Visualize + 3D



Contents

Styles

MODEL:

A mini (3D)

V38532 (1 pose)

Row1_1977711137

A not (3D)

V38532 (1 pose)

Row1_460132198

3 cocryst (3D)

A prep (3D)

Ensemble docking (3D)

Get PDB (3D)

Get PDB from FFC (3D)

No PPrep (CT - DELETED)

Not (CT - DELETED)

PPrep (CT - DELETED)

PPrep - not (CT - DELETED)

Preparation (CT - DELETED)

Thrombin Docking (3D)

Thrombin WaterMap (3D)

TOOLS:

FIT:

Expose workflow parameter or file selection

- Add the label %file% to the Reader node(s) comment or %input% for the Configuration node(s)
- Run Batch test node
 - It analyzes the workflow and prints in the console the arguments to be used on the Admin page
 - The supported reader and configuration nodes can also be checked running:
 - \$SCHRODINGER/run KNIME_batch.py -print allowed
 - The Get help node with the command and argument above
- The corresponding KNIME_batch.py arguments have to be added to LiveDesign admin page
- If you want to control the file(s) and/or parameter(s) from the Parametrize model panel instead of the admin page:
 - On the Admin page change the parameter source from Set fixed to Set default

System specific file configuration location

- If the file rarely has to be changed
 - Change the file from the Admin page
 - eg ADME model
- If the file doesn't change in the project
 - Make a copy of the model for each project (and make it available only to the project)
 - Change the file from the Admin page
 - eg Glide model
- If the reference file has to be changed most of the time
 - Make the model a parametrized model
 - Set default instead of Set fixed
 - eg reference structure in an alignment workflow

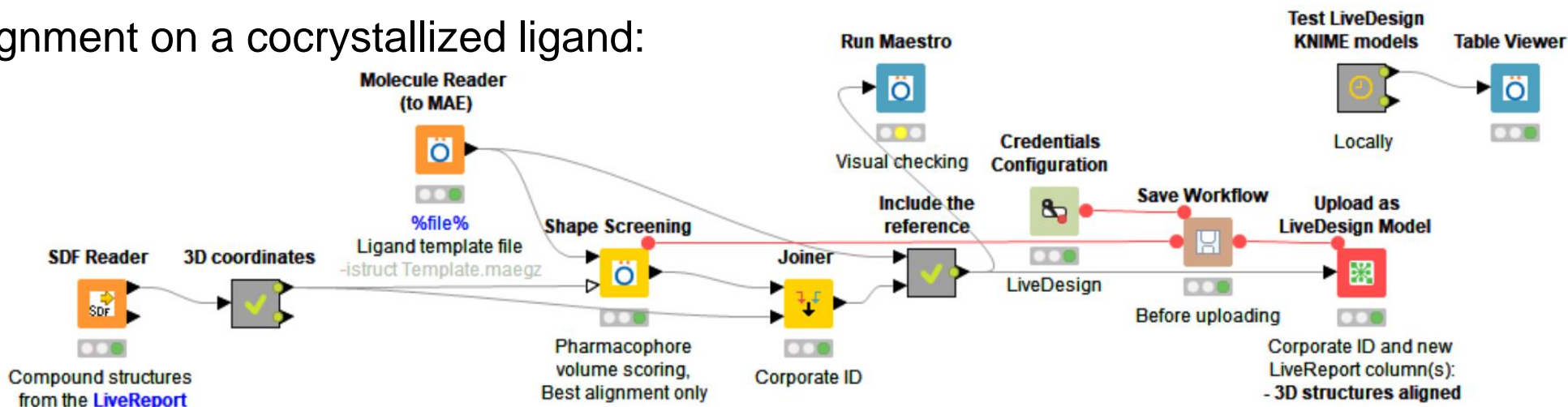
Customize a KNIME computational model

System specific files can be changed from LiveDesign admin page

- No need to open KNIME, nor a the Parameterize model configuration panel each time
- eg ligand alignment template, Glide grid, QSAR model file
- The corresponding KNIME_batch.py arguments have to be added to the page too
 - See Batch test node output

MODEL DATA			
NAME	PARAMETER TYPE	DATA	PARAMETER SOURCE
extra file 1	File	Parent File: <i>mockup.txt</i> Currently: <i>Template.maegz</i> <input type="checkbox"/> Clear Change: <input type="button" value="Browse..."/> No file selected	<input type="text" value="Set Fixed"/>
extra argument 1	Text	<input type="text" value="-istruct Template.maegz"/>	<input type="text" value="Set Fixed"/>

Alignment on a cocrystallized ligand:



System specific files

- If the file rarely has to be changed
 - Change the file from the Admin page
 - eg ADME model
- If the file doesn't change in the project
 - Make a copy of the model for each project (and make it available only to the project)
 - Change the file from the Admin page
 - eg Glide model
- If the reference file has to be changed most of the time
 - Make the model a parametrized model
 - Set default instead of Set fixed
 - eg reference structure in an alignment workflow

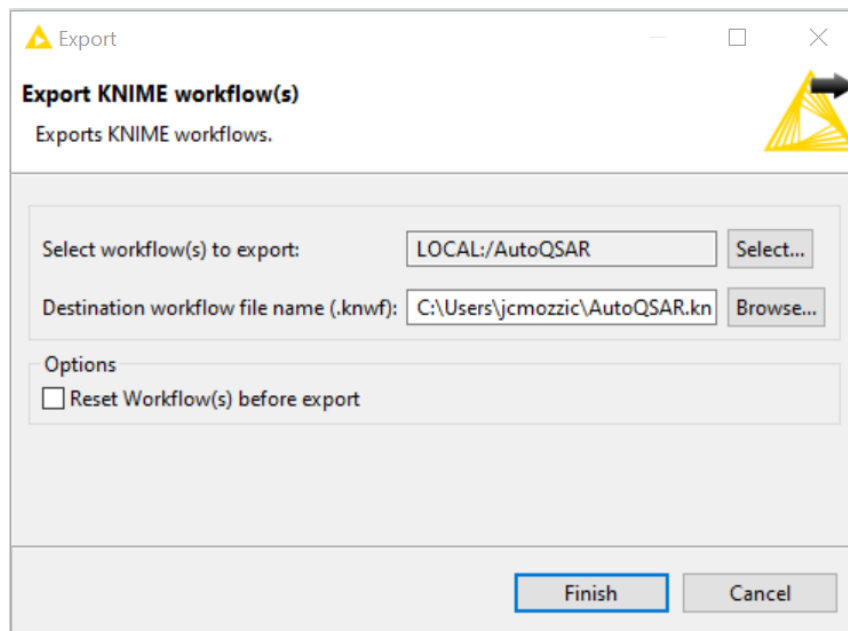
Miscellaneous

KNIME models – open the KNIME workflow

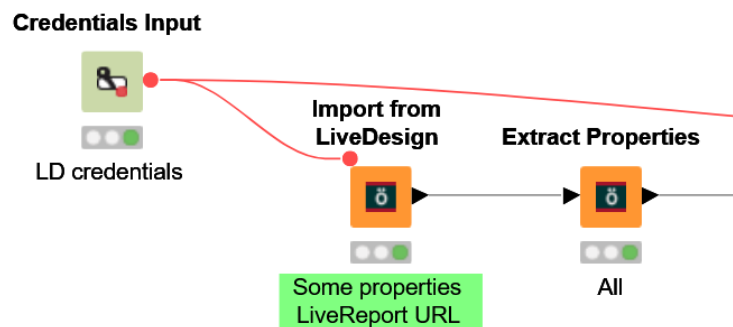
- The KNIME workflow can be downloaded from the model admin page
 - Under Model data, workflow, click on the link to download. Rename the .json file into .knwf
- It is also available on the Tasks admin page
 - In the Input files section, Download
- The workflow executed on the LiveReport data is among the temporary files
 - They are kept if the model failed
 - To inspect these there are ways to make the model to fail intentionally
 - In 21-1 protocols it can be triggered from the model admin page

Save a workflow with the data

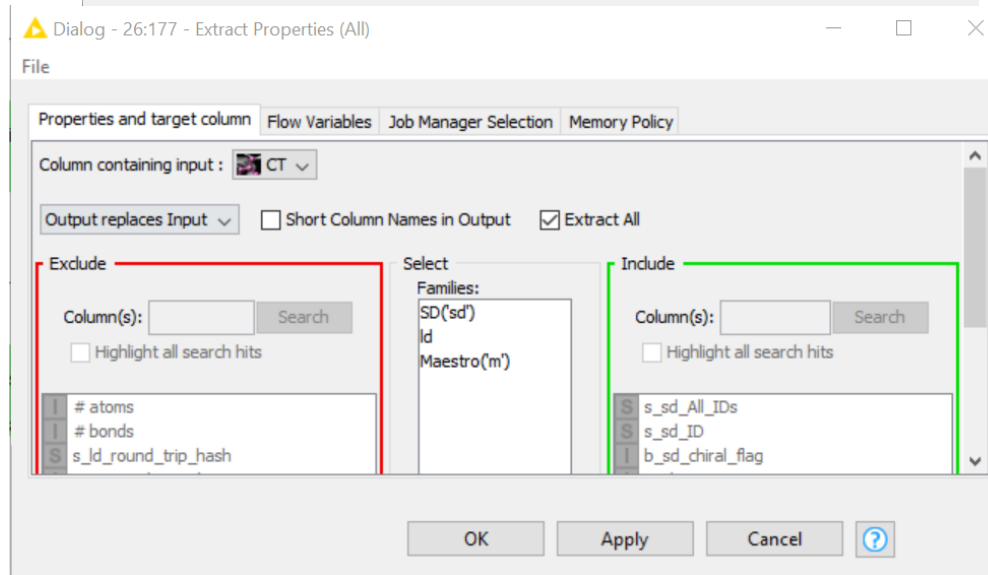
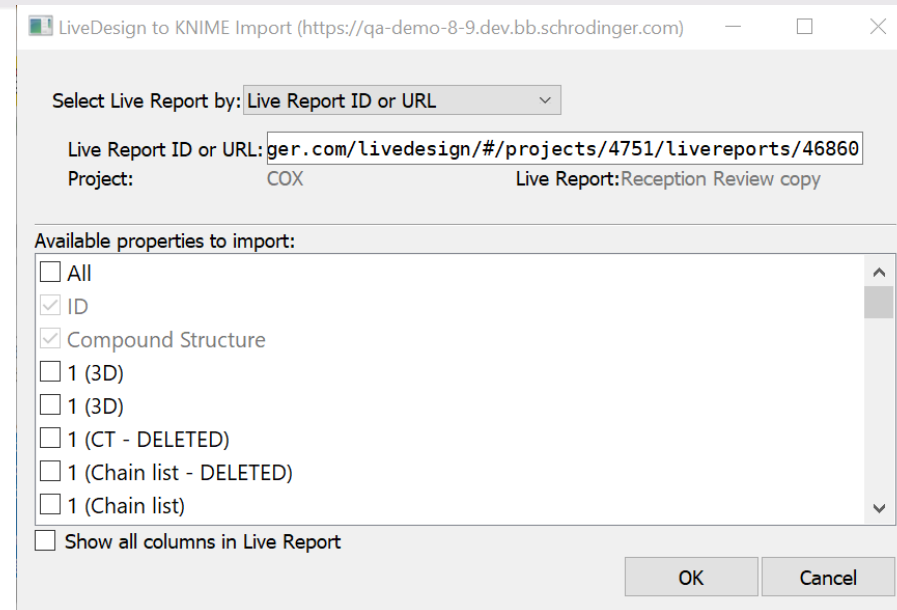
- Under File, export KNIME workflow...
 - Toggle off Reset workflow before export



Extract data from LiveDesign



- Use the Extract data from LiveDesign node
- A credential input node is needed upstream. The variable has to be named LD_credentials
- An Extract properties node downstream populates the output table only with the properties you need immediately. The others are still available in the structure column when needed.
- You can also use the Extract all option in the Extract properties node if you want all the properties to be extracted each time you reexecute the Import from LiveDesign node.



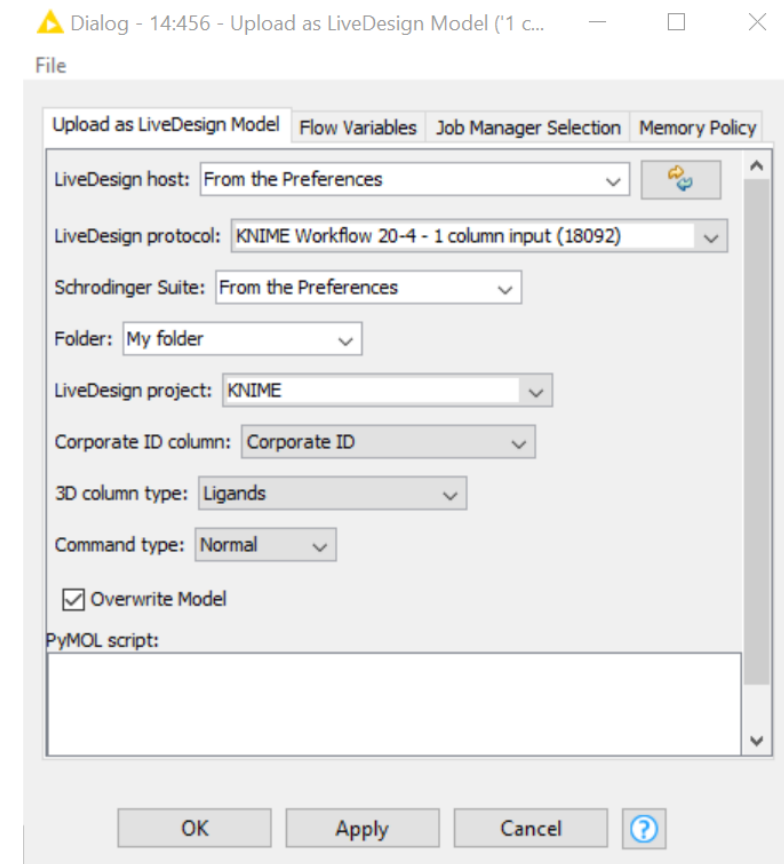
Model organization

- **Folder**

- Empty => Computational Models/KNIME/
- “My folder” => Computational Models/My folder/
- “KNIME/My subfolder” => Computational Models/KNIME/My subfolder
- Same with LiveDesign_admin.py -folder option

- **Project**

- Global : the model is visible
- Project1 : for testing, since only visible in this project
- COX2 : eg QSAR model only relevant to a project



Copy KNIME models

- **Back-up copy of a model or copy for experimenting**
 - In KNIME: open the workflow, change the name and execute the Upload as LiveDesign model node
 - Or in command line: "\$SCHRODINGER"/run -FROM knime LiveDesign_admin.py -type model -mode duplicate -id <ID> -name <New model name> -host <machine> -username <username> (-password <password>)
 - Can also be used for any (non-KNIME) LiveDesign computational models
- **Overwrite a model with a new version**
 - So the results are stored in the same column and previous results can be recalculated automatically
 - In KNIME: open the workflow and execute the Upload as LiveDesign model node
 - Or in command line: "\$SCHRODINGER"/run -FROM knime LiveDesign_admin.py -type model -mode upload -overwrite -file <knwf archive path> -name <New model name> (-folder <folder>)
(-schrodinger <path>)
-host <machine> -username <username> (-password <password>)

Can I update my KNIME model with a more recent Schrodinger Suite version?

- This may work just fine if the model was created a couple of releases ago. However if it is older, some recent protocol improvements may break the forward compatibility. Then you need to switch and use the most recent protocol version.
- From Suite 21-3 you can just deploy the workflow from KNIME GUI running the Upload node again. It will install the latest version of the protocol automatically.
- With an older Suite version you will have to install the protocol first using the workflow: https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Administration/Model%20administration.
- It can also be installed through the command line using LiveDesign_admin.py script. See the help message and the command can also be framed with the Upload protocols node in the workflow above.
 - The KNIME generic protocols are available from \$SCHRODINGER/knime-v*/data/livedesign_protocols and https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Support_materials/Generic_protocols~0AoytFYJTqwyBTrC/



Here are some more resources if you have remaining questions:

https://hub.knime.com/schroedinger/spaces/LiveDesign_models/latest/Support_materials/Documentation/

https://www.schrodinger.com/knimeworkflows/KNIME_newfeatures.pdf

<https://www.schrodinger.com/knime-extensions>

Or contact help@schrodinger.com