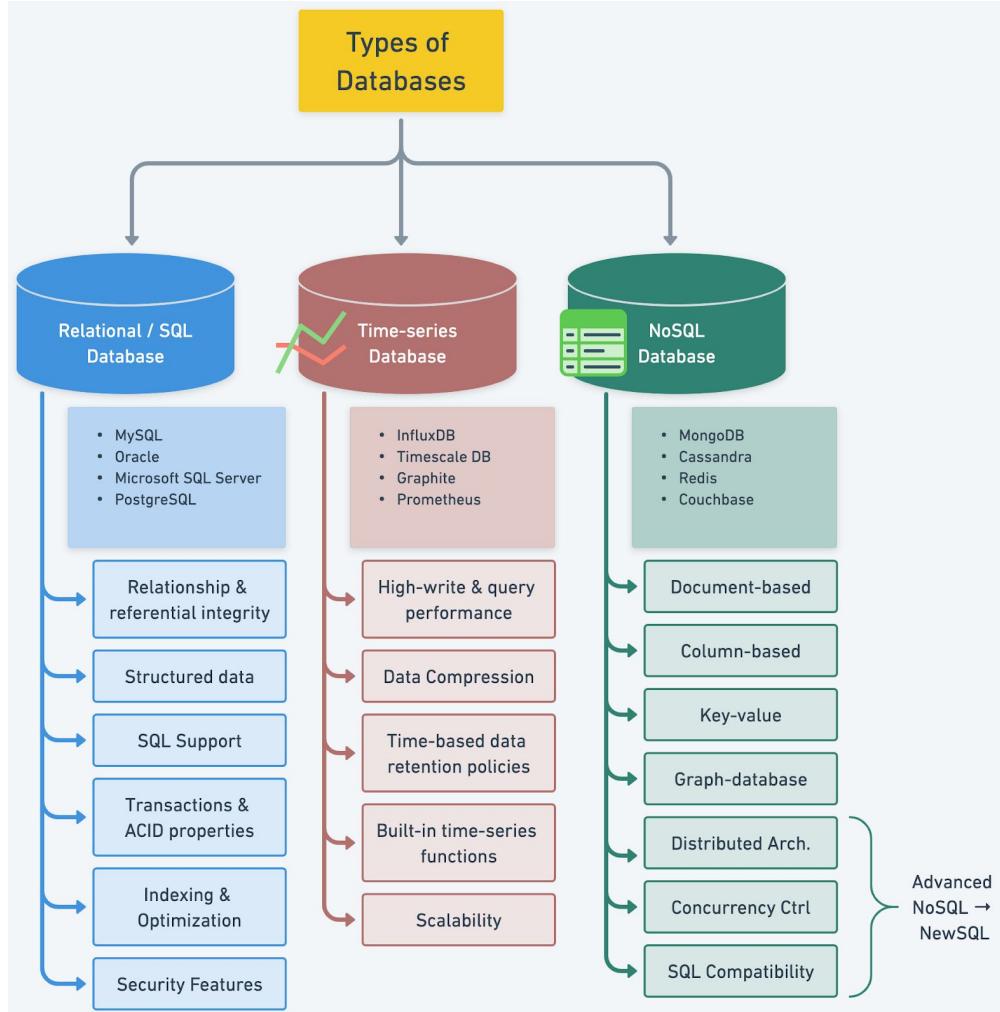


SUMMARY II

(Cli WiSe24/16-16.02-o)

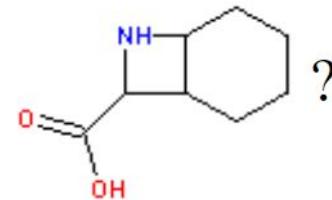
Cartridges, DB's

(Cli WiSe24/08-08.12-x)



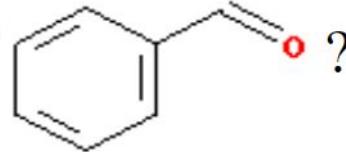
Why a new kind of database?

- Get all the nearest neighbors of



```
SELECT s.Structure FROM Structures s, Neighbors n WHERE  
s.id = n.Neighbor AND n.Compound = 3578;
```

- Get all salts of



```
SELECT s.Structure FROM Structures s, Salts sa WHERE  
s.id = sa.salt AND sa.BaseCompound = 1001;
```

What is a cartridge?

Storing chemical structures and reactions in relational databases with the aid of database extension modules usually called cartridges is an industry standard.

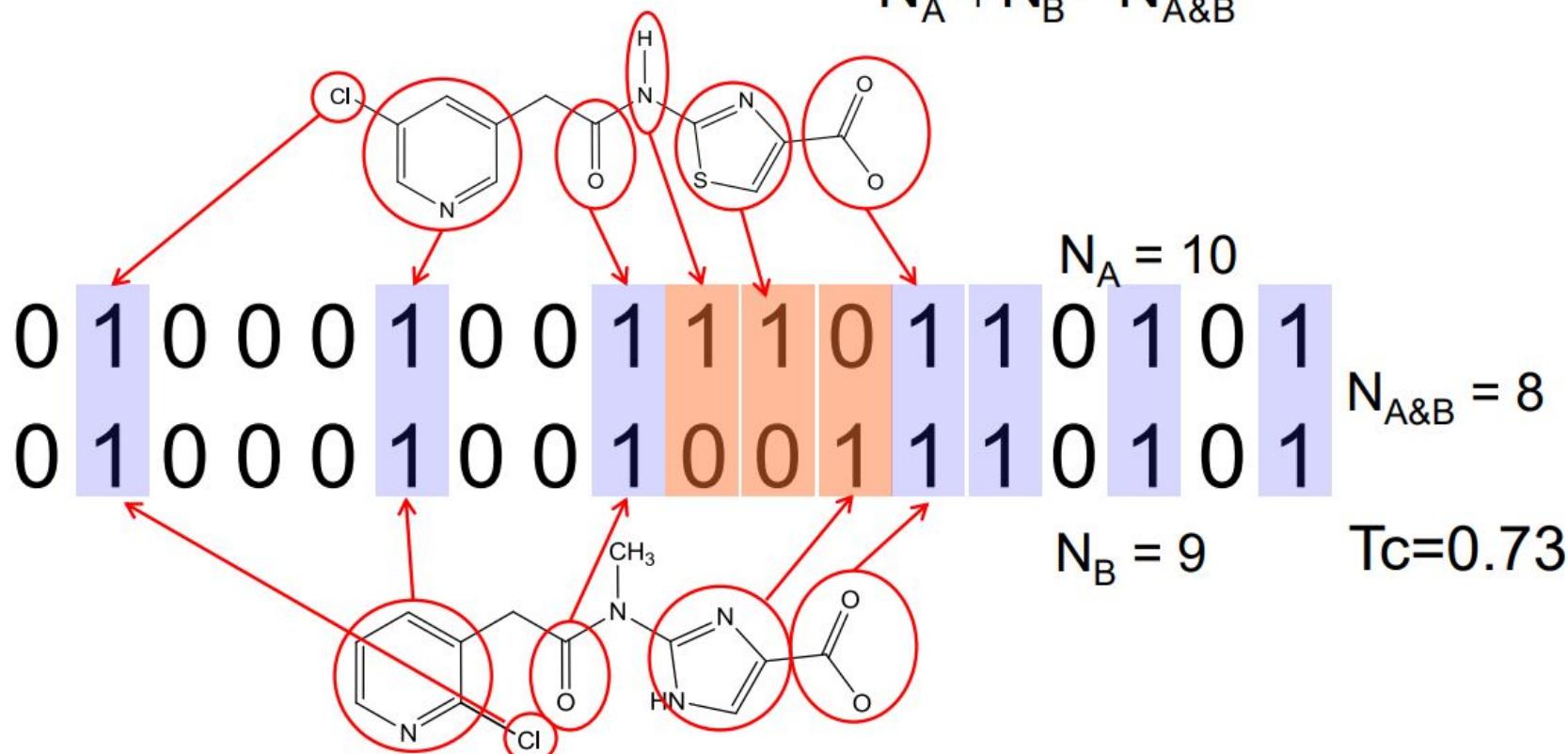
Typically such cartridges provide:

- a fixed set of functions to facilitate structure queries (full-structure, substructure, superstructure, similarity),
- chemistry data file import and export,
- sometimes structure manipulations or standardizations, and
- occasionally a defined set of property computations.

Similarity search

#essential

Tanimoto (Jaccard) coefficient: $Tc = \frac{N_{A\&B}}{N_A + N_B - N_{A\&B}}$



Decision science, Data science, and Knowledge Graphs

(CII WiSe24/09-15.12-x)

Decision science

Overview

Gut feeling - Intuition

Situation	Rationale
Limited Data	If there's insufficient data available to make a statistically valid prediction, a gut feeling based on experience can be a useful tool.
Time Constraint	In situations where a decision needs to be made quickly, and there isn't time to gather and analyze data, relying on gut feeling can be beneficial.
High Complexity	When dealing with highly complex situations where it's difficult to identify or quantify all relevant factors, intuition can play a crucial role.
New or Unique Situations	When encountering a situation for the first time, or if the situation is unique and there's no applicable historical data, gut feeling can guide the decision-making process.
Expertise	If the person has extensive experience and knowledge in a particular field, their gut feeling can be quite accurate and valuable.

Taxonomy of prognostics approaches

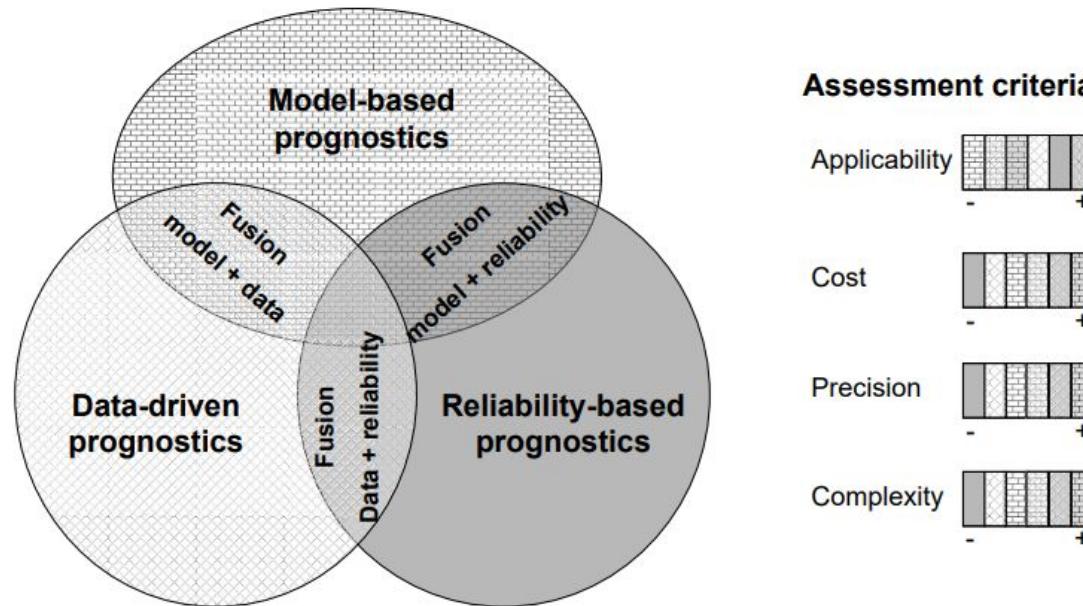
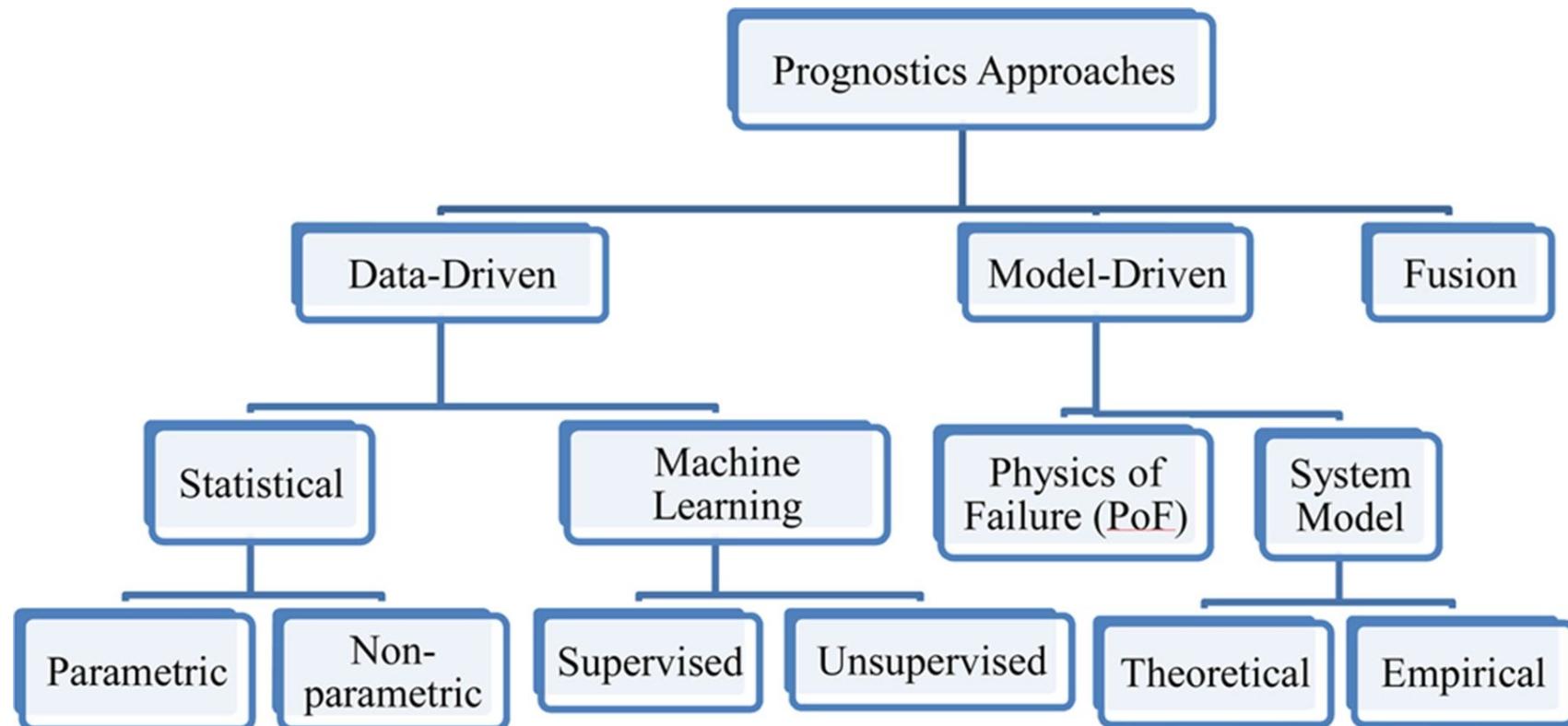


Fig. 4: Main prognostics approaches.

Taxonomy of prognostics approaches



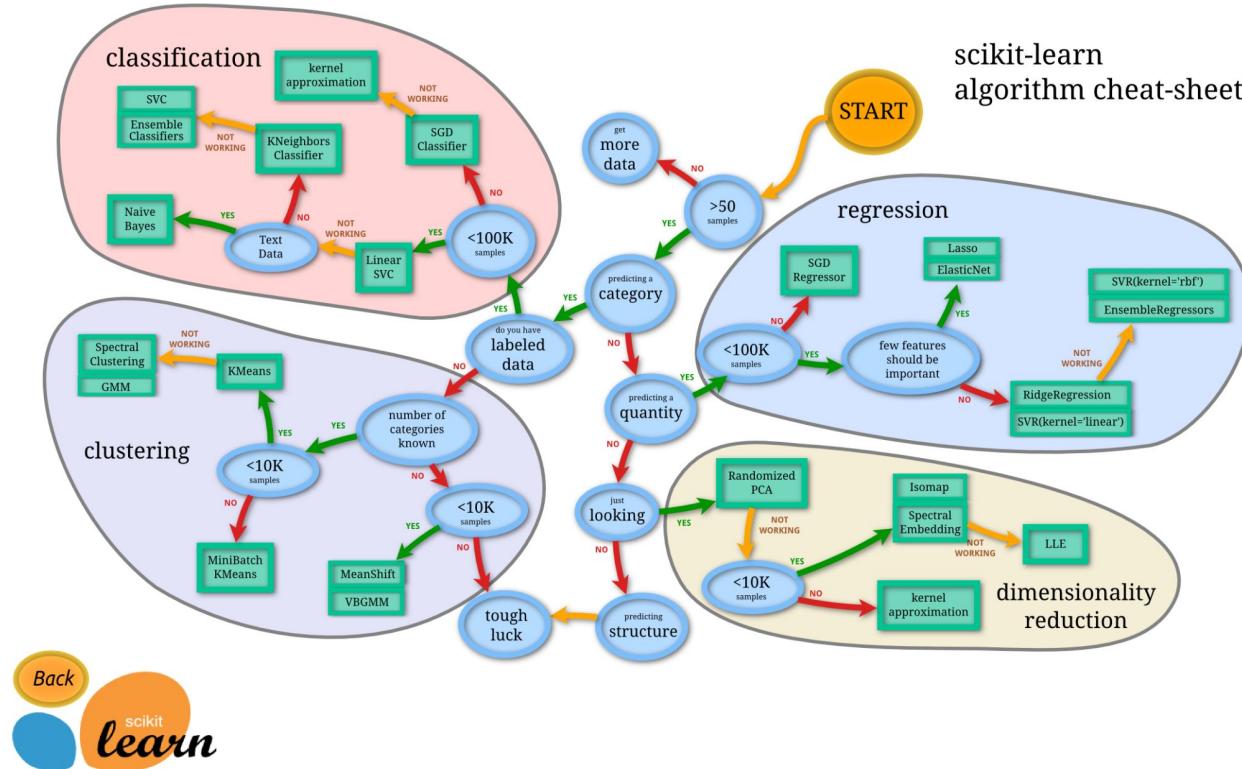
Data driven

7 Steps of Data-Driven Decision-Making

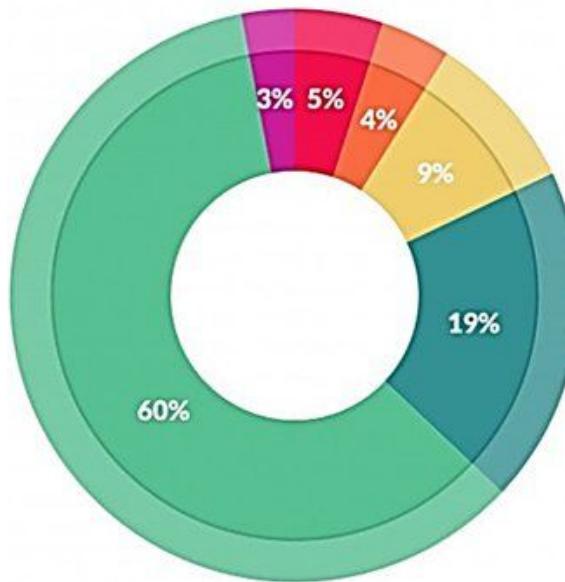
by  BSC Designer



Choosing the right estimator



Data science workflow



- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

<https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-is-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=7e7532a76f63>

Cheminformatics software packages

RDKit is an open-source cheminformatics library that offers functionality for handling chemical data, including reactions. It allows you to import and manipulate reaction data locally within your own scripts or programs.

KNIME is a data analytics platform that includes cheminformatics capabilities. It provides nodes and extensions for working with chemical structures and reactions, allowing you to import and manage reaction data within your local workflows.

ChemAxon offers a suite of cheminformatics tools and software, such as Marvin, JChem, and Reactor. These tools provide options to store and manage reaction data locally, enabling offline access and local storage.

Data science algorithms

Examples

K-NN

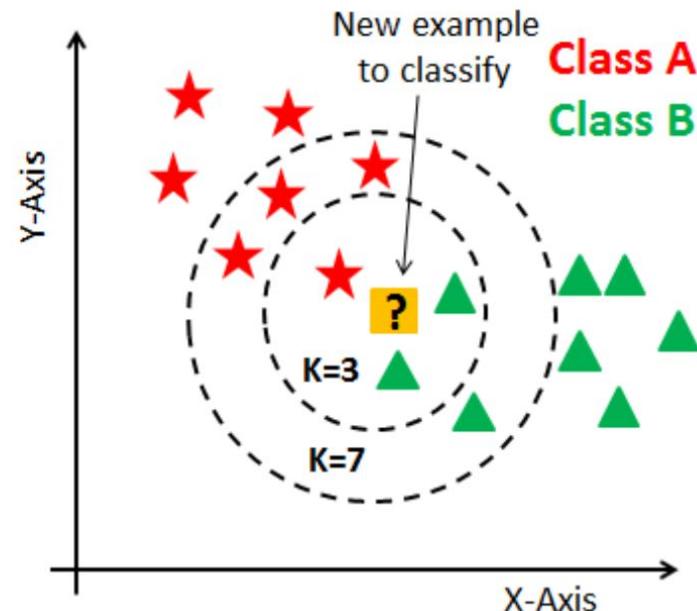
kNN is one of the simplest yet powerful supervised ML algorithms.

It is widely used for classification problems as well as can be used for regression problems.

The data-point is classified on the basis of its k Nearest Neighbors, followed by the majority vote of those nearest neighbors; a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

Example:

https://colab.research.google.com/drive/1zK9jBGzWOvJVnh9iminYVRqG0Zchhn_n#scrollTo=zEH7yf6kLYBy



"Introduction To K-Nearest Neighbors (Knn) Algorithm". 2021. Medium.

<https://ai.plainenglish.io/introduction-to-k-nearest-neighbors-knn-algorithm-e8617a448fa8>

Pipelines

Data Pipeline

A Data Pipeline is a term used to describe a workflow consisting of one or more tasks that ingest, move, and transform raw data from one or more sources to destination. Usually, the destination data is then used for analysis or other business functions.

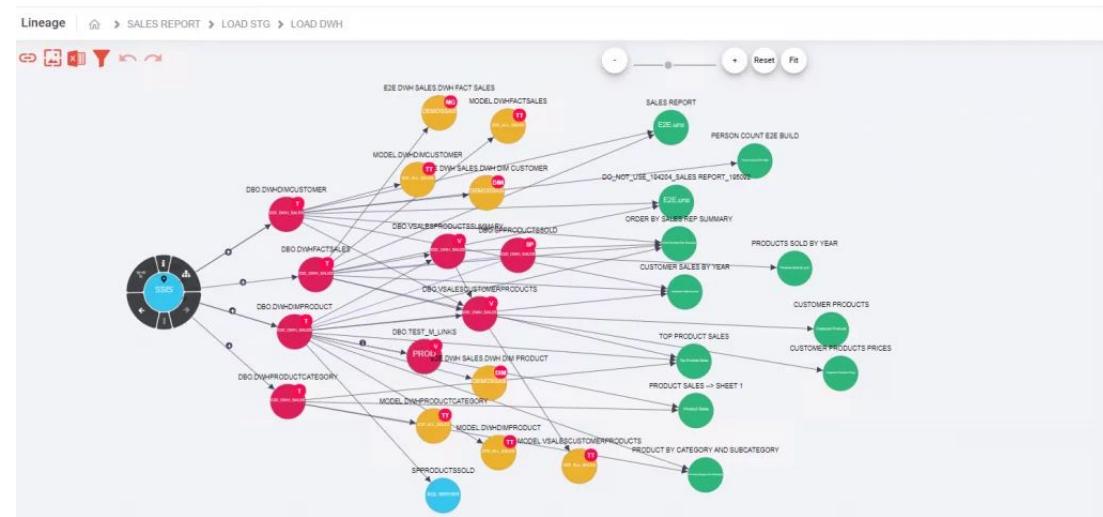
Data Lineage

Data lineage includes:

- the data origin,
- what happens to it, and
- where it moves over time.

Data lineage gives visibility while greatly simplifying the ability to trace errors back to the root cause in a data analytics

Provenance is defined as a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing.



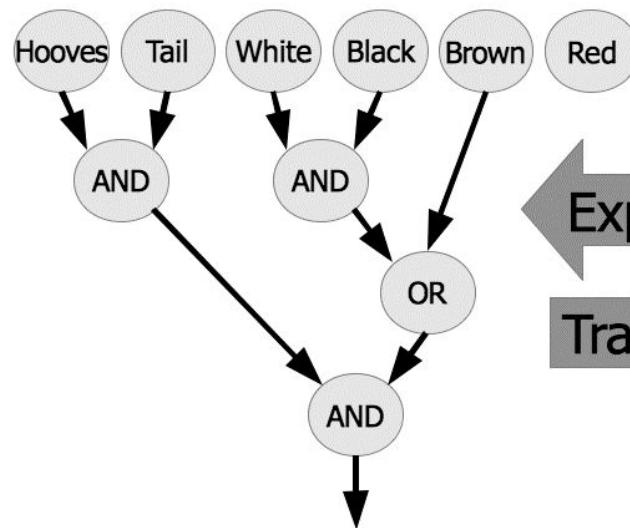
https://en.wikipedia.org/wiki/Data_lineage

<https://www.octopai.com/>

<https://getmanta.com/data-lineage-done-right-demo>

Knowledge representation

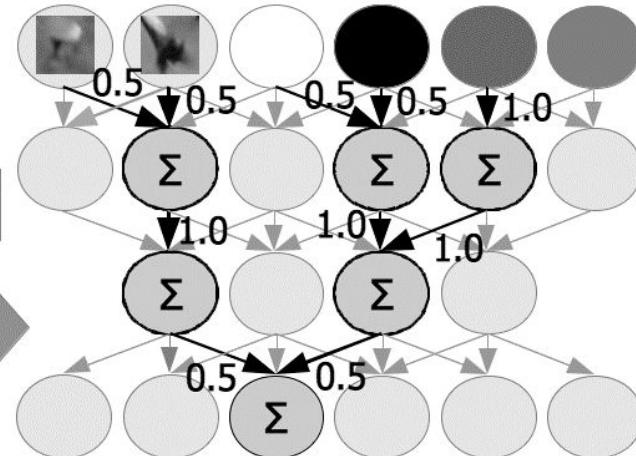
With focus on KG's



(Hooves AND Tail) AND
((White and Black) OR Brown)

=> Horse

Explain
Transfer



2. Ontology

An Ontology for process chemistry would define classes, properties, and relationships specific to the domain. Here's an example:

Classes:

- Reaction
- Catalyst
- Reactant
- Product

Properties:

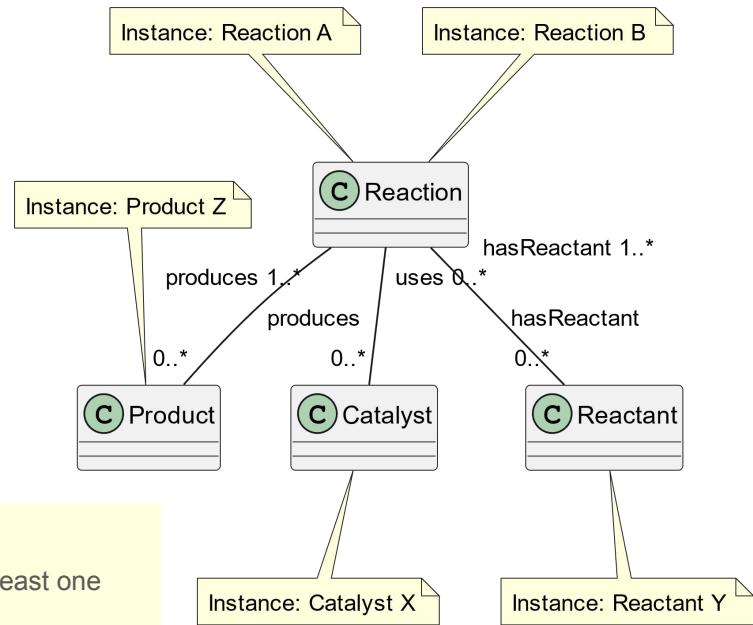
- uses (relationship between Reaction and Catalyst)
- hasReactant (relationship between Reaction and Reactant)
- produces (relationship between Reaction and Product)

Axioms/Constraints:

- Every Reaction must have at least one Reactant.
- Every Reaction must produce at least one Product.
- A Catalyst may be used in multiple Reactions.

Instances:

- Reaction A, Reaction B (instances of Reaction)
- Catalyst X (instance of Catalyst)
- Reactant Y (instance of Reactant)
- Product Z (instance of Product)



3. (Abstract) Knowledge Graph

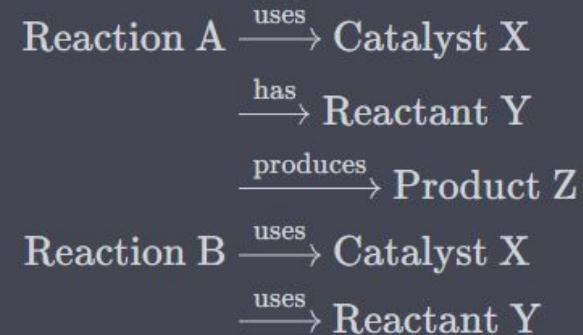
A Knowledge Graph can represent entities and relationships within the domain of process chemistry. Below is a visual representation and description of a small Knowledge Graph:

Entities:

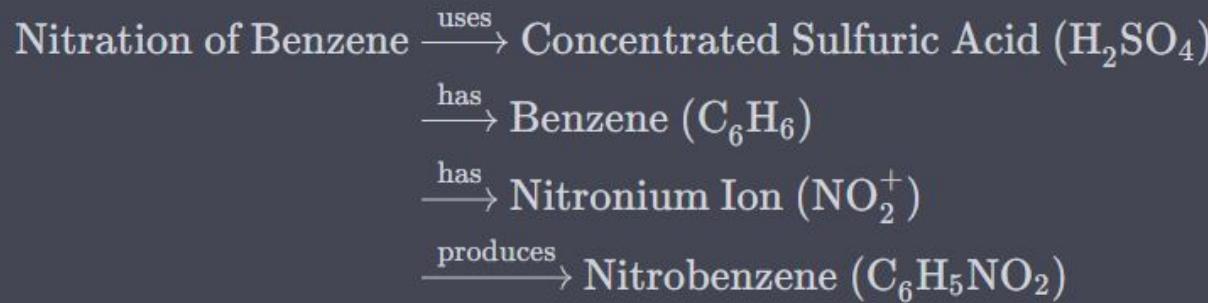
- Reaction A
- Reaction B
- Catalyst X
- Reactant Y
- Product Z

Relationships:

- Reaction A uses Catalyst X
- Reaction B uses Catalyst X
- Reaction A has Reactant Y
- Reaction A produces Product Z
- Reaction B uses Reactant Y



4. Real-live Knowledge Graph



Entities:

- Reaction: Nitration of Benzene
- Catalyst: Concentrated Sulfuric Acid (H₂SO₄)
- Reactant: Benzene (C₆H₆)
- Intermediate: Nitronium Ion (NO₂⁺)
- Product: Nitrobenzene (C₆H₅NO₂)

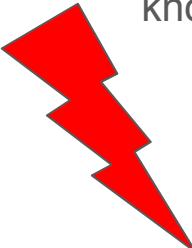
Relationships:

- Nitration of Benzene uses Concentrated Sulfuric Acid
- Nitration of Benzene has Reactant Benzene
- Nitration of Benzene has Intermediate Nitronium Ion
- Nitration of Benzene produces Nitrobenzene

Interaction between Knowledge Graph and the Data layer

A knowledge graph serves as a semantic layer that represents relationships between entities in a structured format. It is on top of the data layer, which contains raw data stored in databases or other storage mechanisms. Queries from the knowledge graph to the data layer generally follow these steps:

1. **Semantic Interpretation:** A user query is first interpreted in the context of the knowledge graph to identify relevant entities and relationships.
2. **Query Translation:** The interpreted query is then translated into a format compatible with the underlying data layer, typically SQL, SPARQL, or API calls.
3. **Data Retrieval:** The translated query is executed in the data layer, retrieving the relevant data points.
4. **Result Mapping:** The retrieved data is mapped back to the entities and relationships in the knowledge graph for presentation or further analysis.



```
SELECT * FROM Reactions
WHERE Reactant = 'Benzene'
AND Catalyst = 'Concentrated Sulfuric Acid';
```

Reasoning

Inference in Knowledge Graph

In a well-designed knowledge graph, inference rules can be implemented to derive new information from existing data.

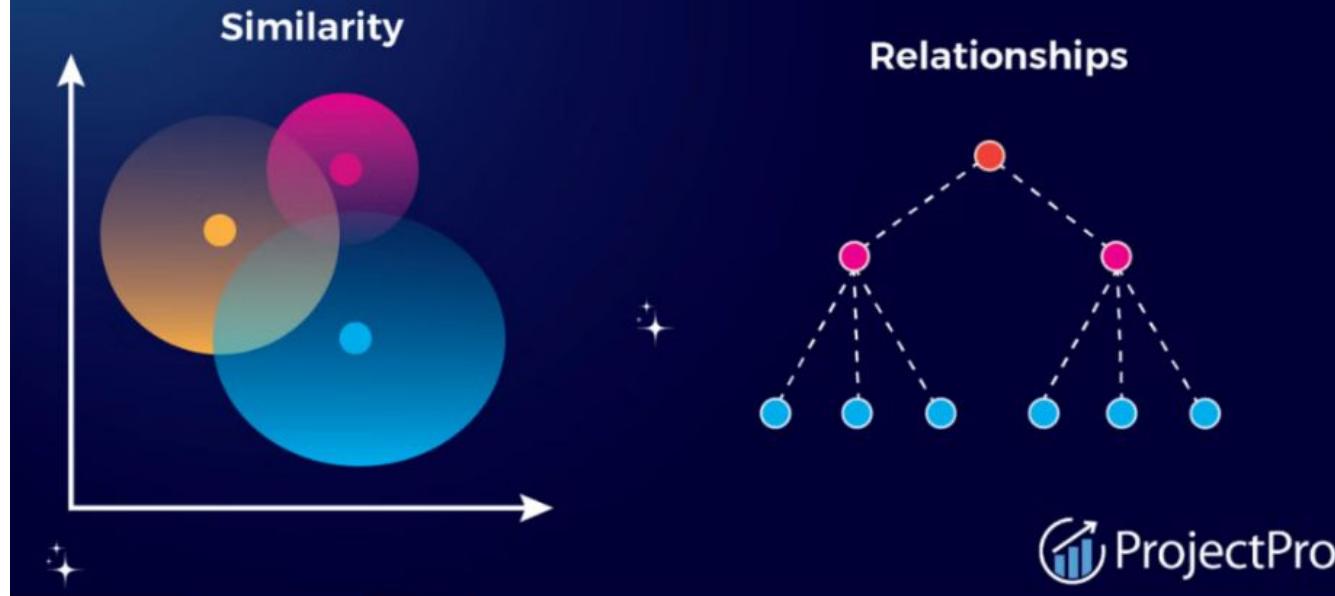
- **Rule Implementation:** An inference rule might be: "If a reaction 'uses' Concentrated Sulfuric Acid, then the reaction 'is acidic'."
- **Application:** The rule automatically infers that "Nitration of Benzene 'is acidic'" based on the original relationship stating that Nitration of Benzene "uses" Concentrated Sulfuric Acid.

Query with Inference:

To query all acidic reactions, use a more generalized query:

```
SELECT * FROM Reactions WHERE ReactionProperty = 'acidic';
```

Vector Database vs Graph Database

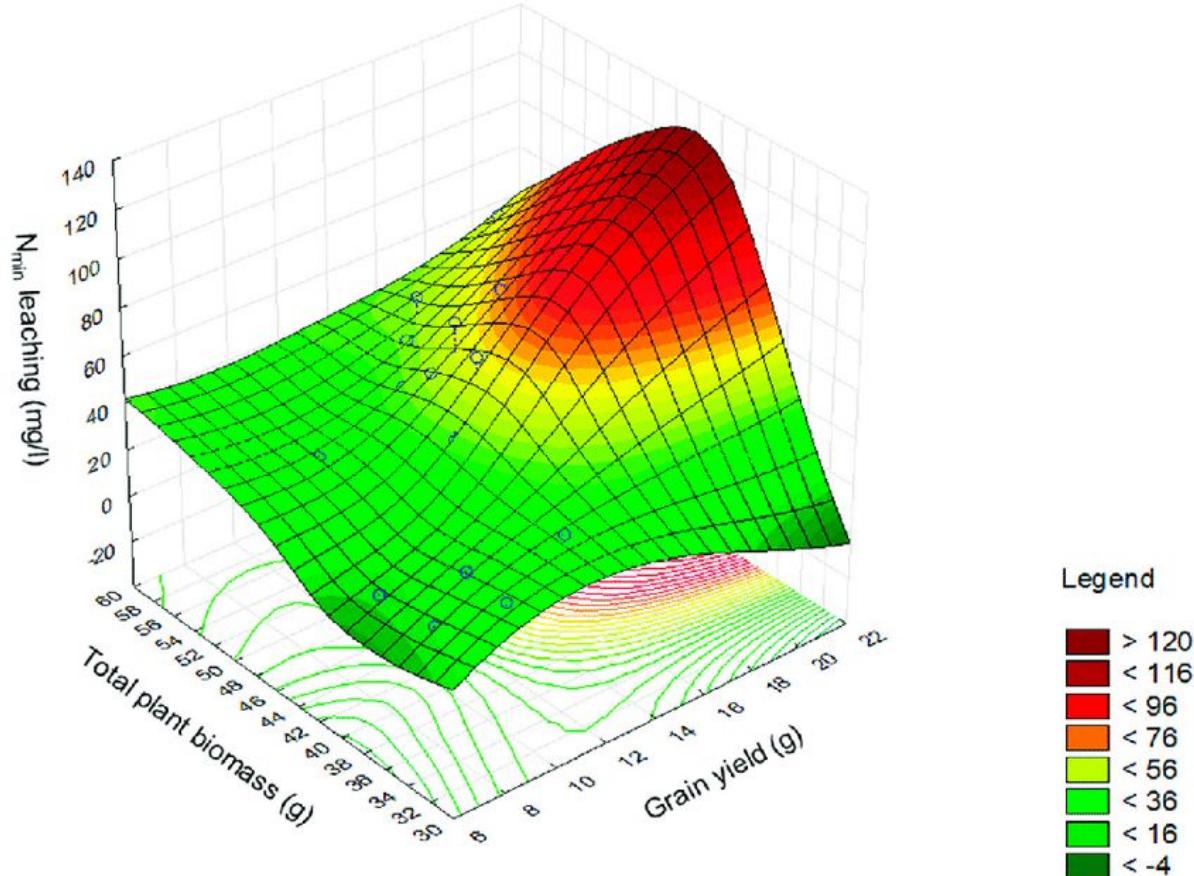


DoE and Analytics

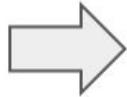
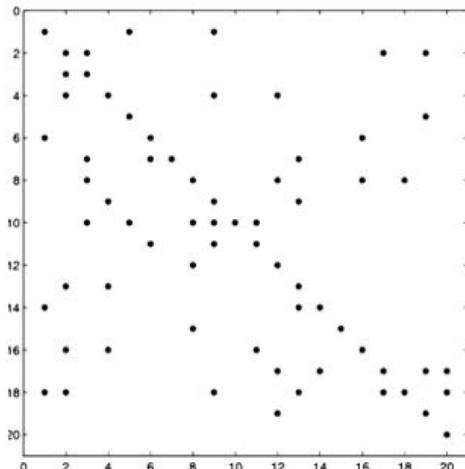
(Cli WiSe24/10-22.12-x)

Observable space

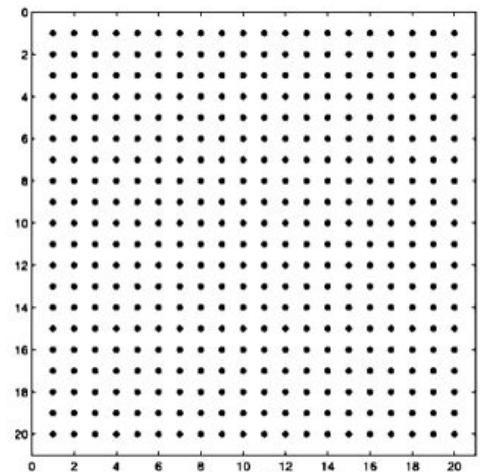
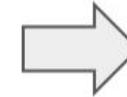
Reaction space



Observable space



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4				X						
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										



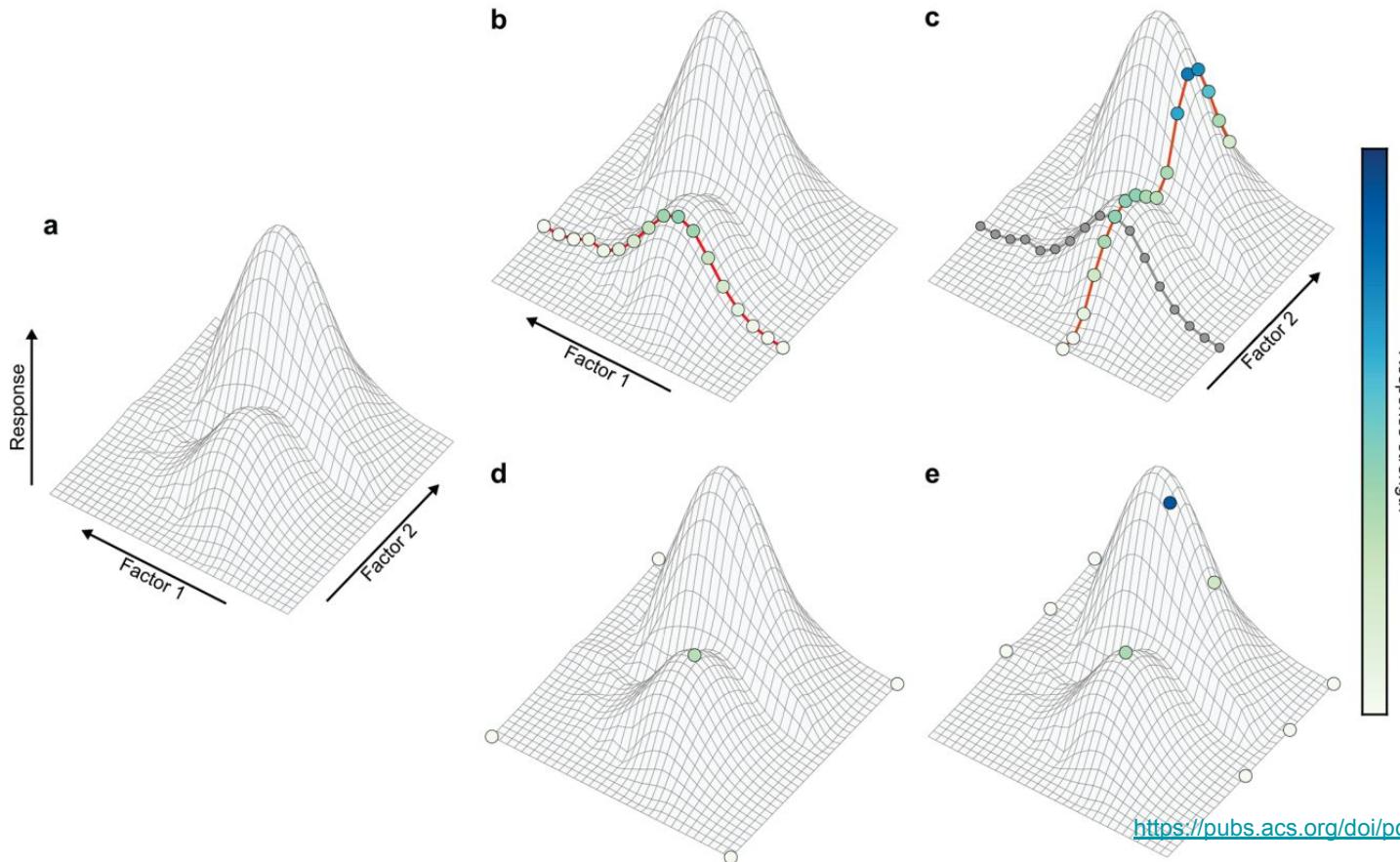
(Gressling)

DoE

<https://www.prismtc.co.uk/resources/blogs-and-articles/article-the-sequential-nature-of-classical-design-of-experiments>

DoE

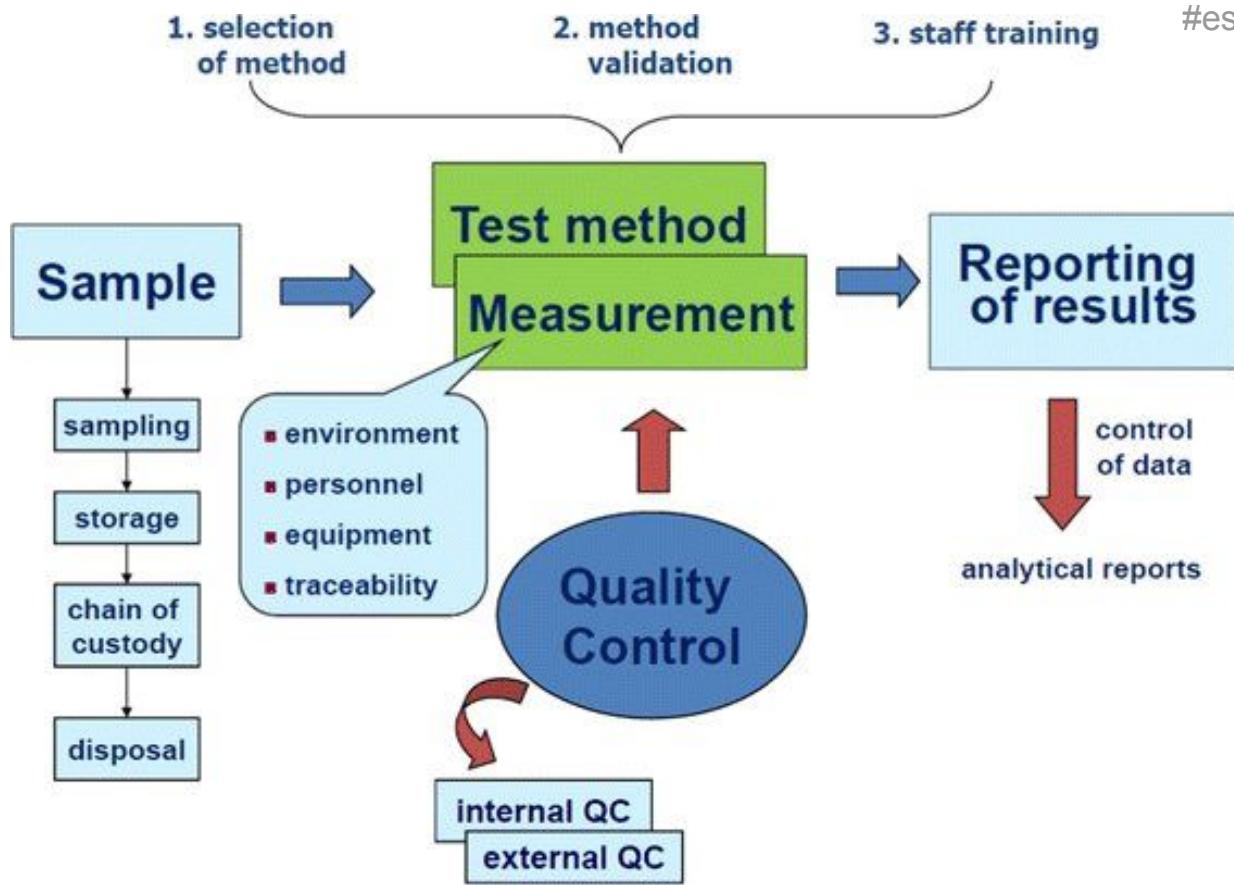
. Exploration of a response surface (a) using OFAT (b,c) or DoE (d,e)



Analytic process

Laboratory

Testing

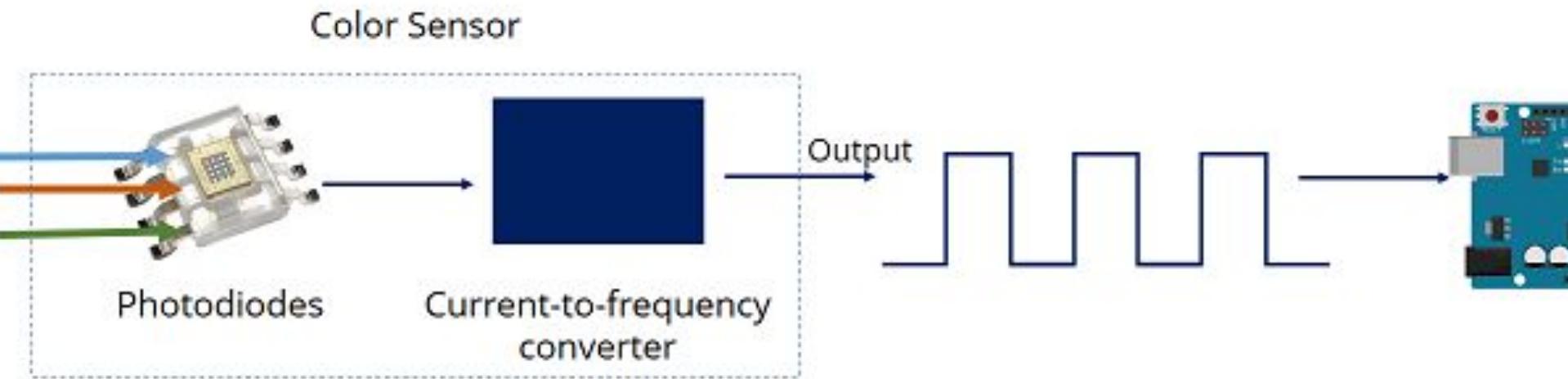


Risk-based thinking for chemical testing, Siu-kay Wong

https://www.researchgate.net/publication/314109558_Risk-based_thinking_for_chemical_testing

Color sensor (TCS3200D)

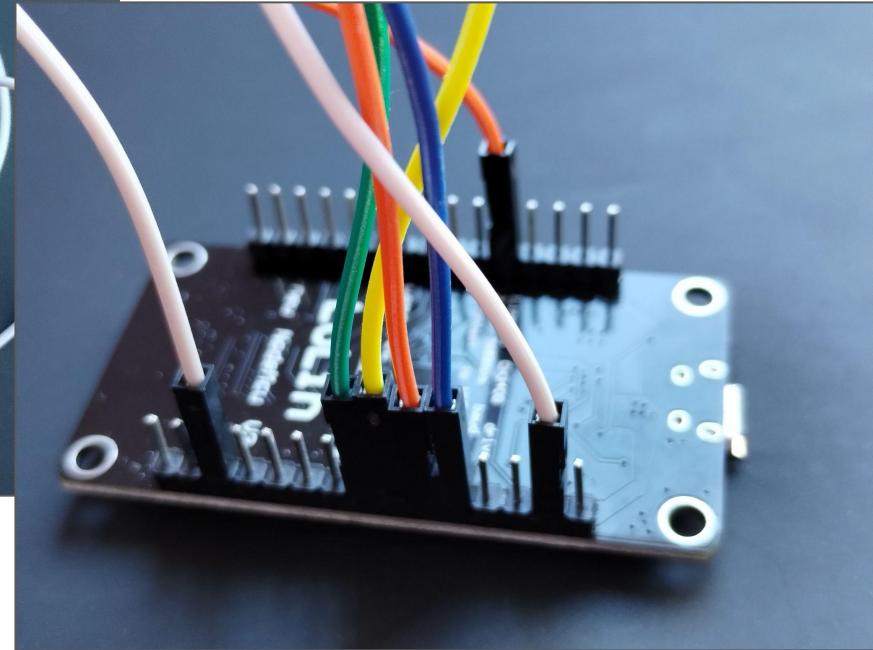
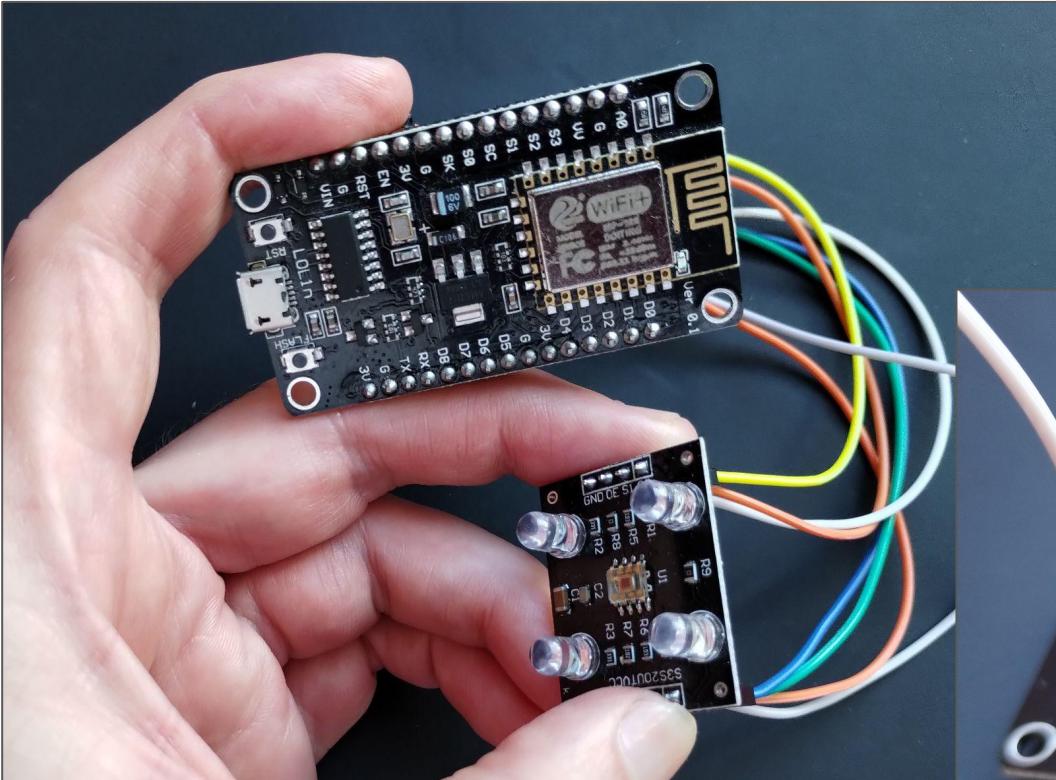




Sana, Rui Santos, Damian Garcia, Sara Santos, Duncan, Rob, Anant Chandak, et al. 2019. "Arduino Color Sensor TCS230 TCS3200." Random Nerd Tutorials. April 2.

<https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/>.

#essential



Code (<https://chat.openai.com/share/f43c83f5-e32e-4cc9-9a1f-111c564e30c8>)

```
#include <Arduino.h>

// Define the pins connected to the TCS3200
#define S0_PIN D6
#define S1_PIN D7
#define S2_PIN D5
#define S3_PIN D8
#define OUT_PIN D2

void setup() {
    Serial.begin(115200);

    pinMode(S0_PIN, OUTPUT);
    pinMode(S1_PIN, OUTPUT);
    pinMode(S2_PIN, OUTPUT);
    pinMode(S3_PIN, OUTPUT);
    pinMode(OUT_PIN, INPUT);

    // Set frequency scaling to 20%
    digitalWrite(S0_PIN, HIGH);
    digitalWrite(S1_PIN, LOW);
}

void loop() {
    long red, green, blue;

    // Measure red
    digitalWrite(S2_PIN, LOW);
    digitalWrite(S3_PIN, LOW);
    red = pulseIn(OUT_PIN, LOW);

    // Measure green
    digitalWrite(S2_PIN, HIGH);
    digitalWrite(S3_PIN, HIGH);
    green = pulseIn(OUT_PIN, LOW);

    // Measure blue
    digitalWrite(S2_PIN, LOW);
    digitalWrite(S3_PIN, HIGH);
    blue = pulseIn(OUT_PIN, LOW);

    Serial.print("Red: ");
    Serial.print(red);
    Serial.print(" Green: ");
    Serial.print(green);
    Serial.print(" Blue: ");
    Serial.println(blue);

    delay(500); // Delay before next reading
}
```

Code for the serial monitor

The screenshot shows a code editor window with a dark theme. At the top, there's a tab labeled "serialReader.py" with a save icon, followed by "U" and a close button. Below the tabs, the code is displayed:

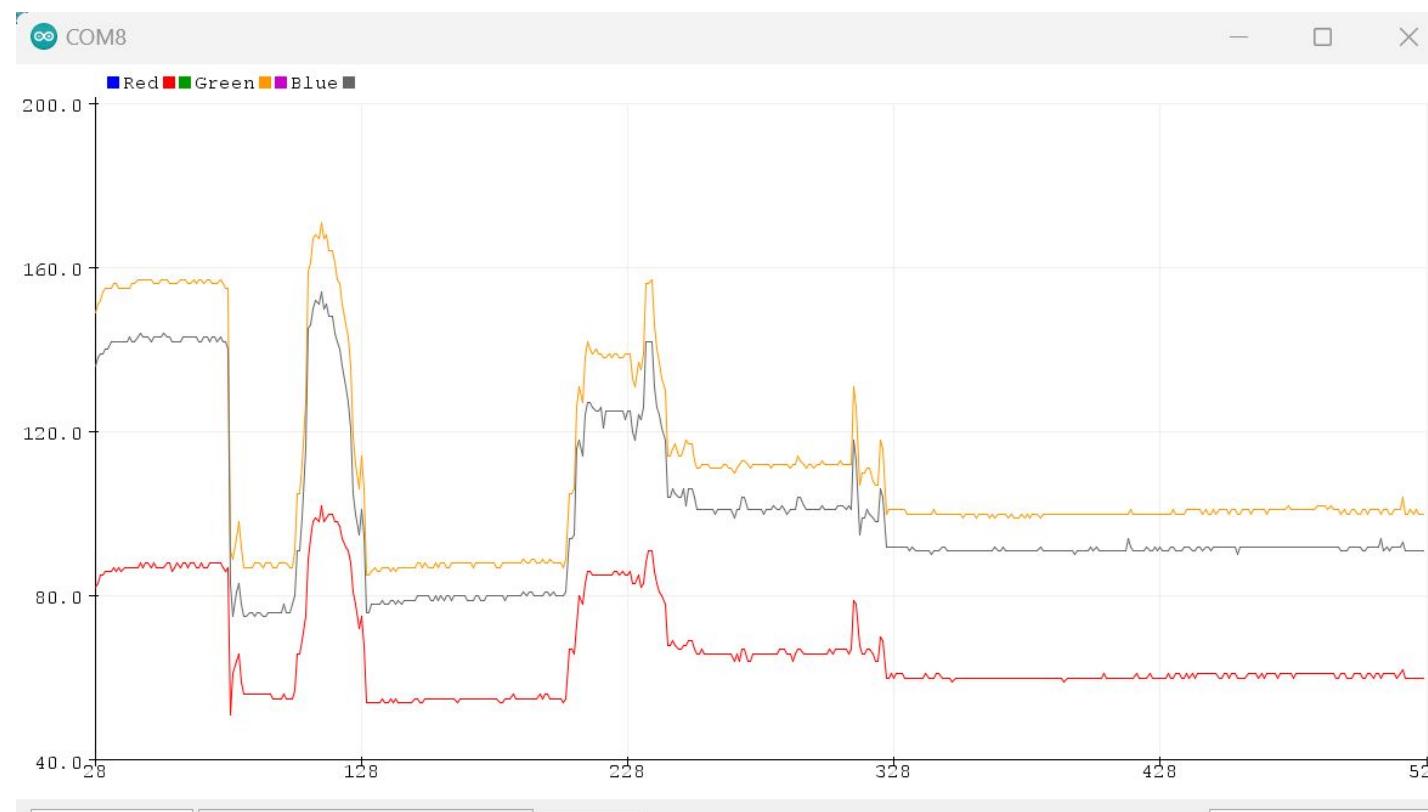
```
serialReader > serialReader.py > ...
5  # Set up serial connection (adjust the COM port and baud rate according
6  ser = serial.Serial('COM4', 115200, timeout=1)
7
8  # Define the CSV file name
9  # Generate a timestamp
10 timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
11
12 # Define the CSV file name with the timestamp
13 csv_file = f"serialReader\\data\\arduino_data_{timestamp}.csv"
14
15 try:
```

Below the code editor, there's a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), PORTS, and SERIAL MONITOR.

The "TERMINAL" tab displays the following text, which is also highlighted in the code editor:

```
2023-11-12 14:21:36, 25,59,53
2023-11-12 14:21:36, 22,50,44
2023-11-12 14:21:37, 22,50,45
2023-11-12 14:21:37, 22,51,45
2023-11-12 14:21:38, 22,51,47
2023-11-12 14:21:38, 22,50,45
2023-11-12 14:21:39, 22,50,45
2023-11-12 14:21:39, 22,51,45
2023-11-12 14:21:40, 23,51,45
2023-11-12 14:21:40, 23,51,47
2023-11-12 14:21:41, 23,53,47
```

Result

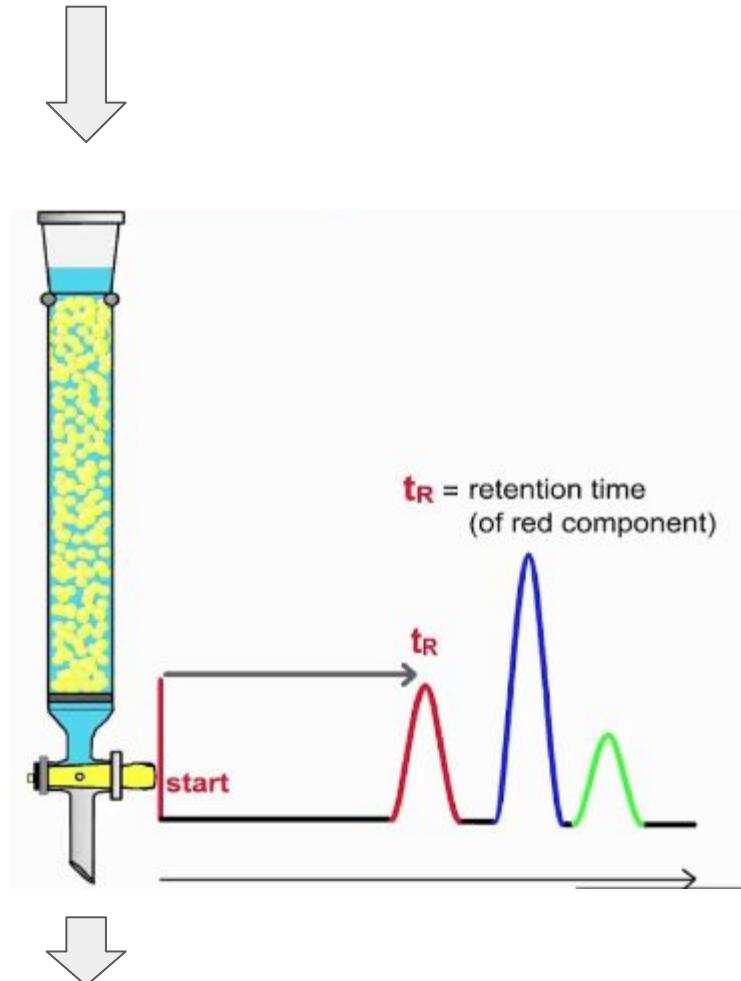


Analytical method development

Method

Everything you need to ***objectively repeat*** the measurement:

- Eluent
- Solid phase
- Pressure
- ...



Calibration

Calibration

Calibration may be required for the following reasons:

- a new instrument
- after an instrument has been repaired or modified
- moving from one location to another location
- when a specified time period has elapsed
- when a specified usage (operating hours) has elapsed
- before and/or after a critical measurement
- after an event, for example
- after an instrument has been exposed to a shock, vibration, or physical damage, which might potentially have compromised the integrity of its calibration
- sudden changes in weather
- whenever observations appear questionable or instrument indications do not match the output of surrogate instruments
- as specified by a requirement, e.g., customer specification, instrument manufacturer recommendation.

<https://en.wikipedia.org/wiki/Calibration>

Calibration

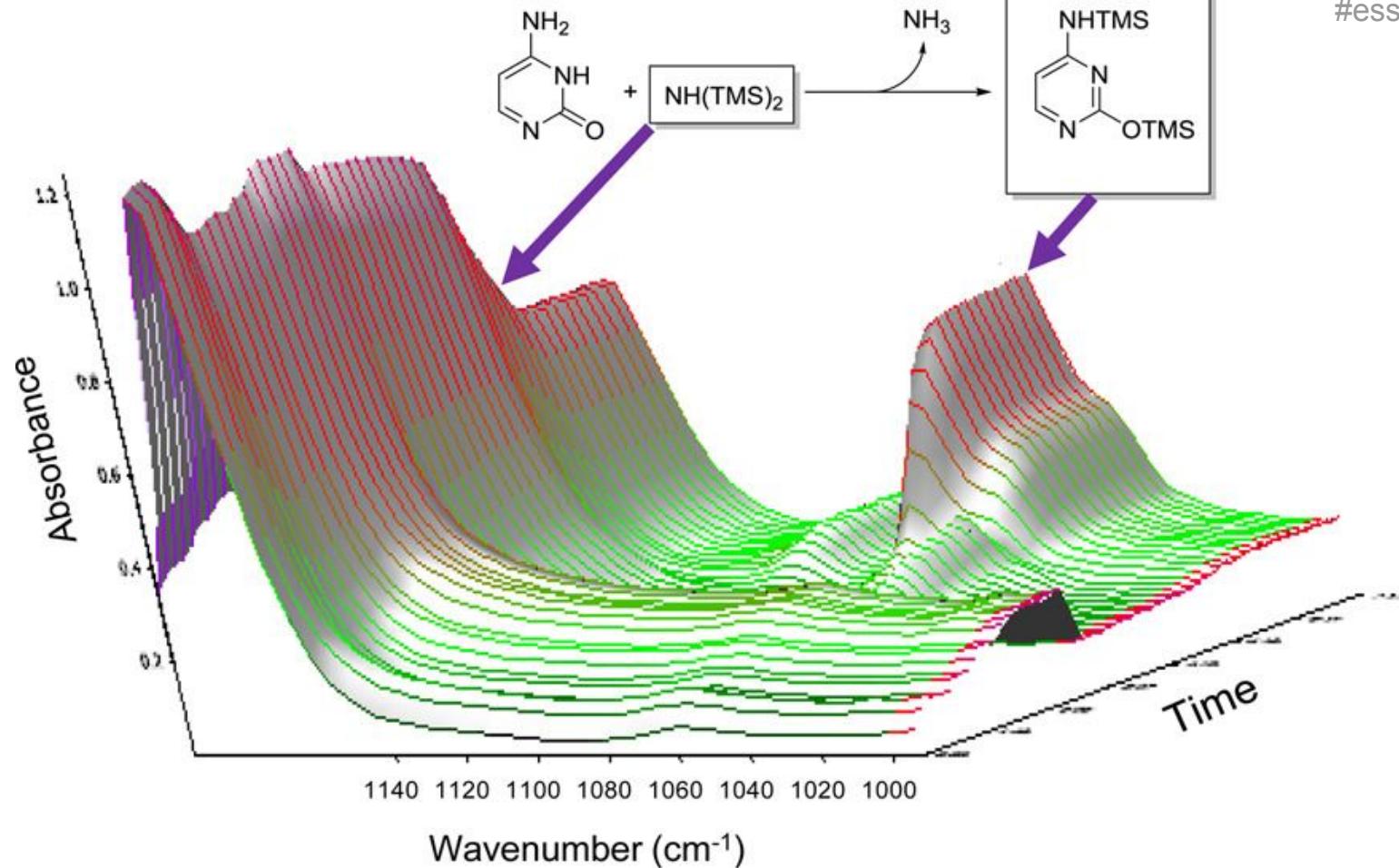
Measurement Data Sheet Thermometer Calibration					
Owner :	Date Received:				
Instrument Description:	Date of Calibration:				
Brand :	Due Date :				
Model/type:	Range:				
Serial no:	Resolution :				
Calibration Method:	Loc. of Equipment:				
Ambient Temp. :					
Relative Humidity:					
Reference Standard Used:					
Calibration Results:					
Applied Temperature	Unit Under Calibration(UUC) Reading			Average	Correction
	Trial 1	Trial 2	Trial 3		
Remarks					
Calibrated By:					

<https://calibrationawareness.com/elements-in-implementing-international-calibration-laboratory>

PAT



PAT



Integration with chatGPT

In [7]:

```
1 import pandas as pd
2 from pandasai import PandasAI
3
4 # getBatch Data
5 batch = getS88("THGR_0044-123")
6 trend = {
7     'Unit': batch.get('Unit'),
8     'StartTimeUTC': batch.get('StartTimeUTC'),
9     'Data': [[v.get('T'), v.text] for v in batch.findall('V')]}
10 }
11 df = pd.DataFrame(trend.toPandas())
12
13 # Instantiate a LLM
14 from pandasai.llm.openai import OpenAIBAY
15 llm = OpenAI(api_token="sk-xyXXXXXXXXXXXXnL82b4")
16
17 # Interpretation
18 pandas_ai = PandasAI(llm)
19 pandas_ai.run(df, prompt='What are the important landmarks in this trends?')
```

Applied Thermodynamics

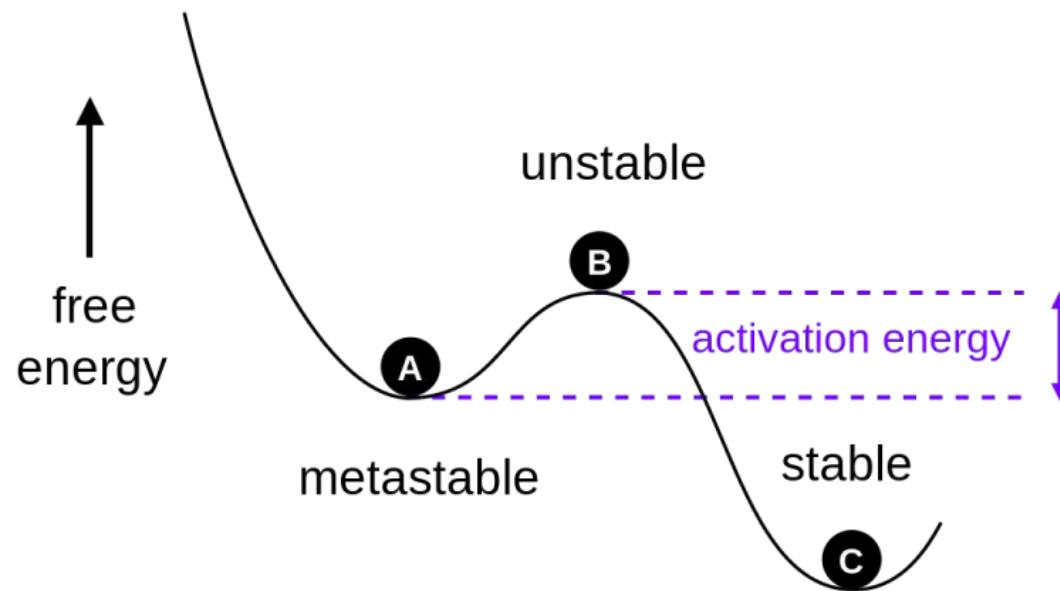
Recap: Thermodynamics and kinetics

“Thermodynamic stability refers to the status of *products*, whereas kinetic stability refers to the status of *reactants*”

“Thermodynamic stability is the stability of the lowest energy state of a system while kinetic stability is the stability of the highest energy state of a system”

“Thermodynamics describes the overall properties, behavior, and equilibrium composition of a system; kinetics describes the rate at which a particular process will occur and the pathway by which it will occur.”

Recap: Thermodynamics



Cantera for thermodynamic properties

Cantera is a suite of tools for solving problems involving chemical kinetics, thermodynamics, and transport processes. Solution() and Kinetics() objects provide all of the methods necessary to compute the thermodynamic quantities associated with each reaction, reaction rates, and species creation and destruction rates.

- <https://cantera.org/documentation/index.html>
- <https://cantera.org/tutorials/python-tutorial.html>
- <https://cantera.org/examples/python/>
- [6b-2. Cantera introduction](#)



Cantera includes the NASA thermochemical database [GM71, MGaMAR93], which contains fits for a large number of gases, including argon, helium, and monatomic oxygen.

Statistical thermodynamics example: Monoatomic gases

An ideal gas is a theoretical gas composed of many randomly moving point particles that are not subject to interparticle interactions.

The ideal gas concept is useful because it obeys the ideal gas law, a simplified equation of state, and is amenable to analysis under statistical mechanics.

$$PV = nRT$$

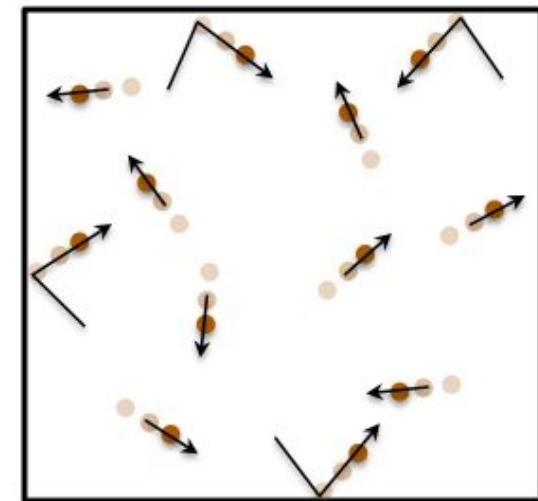
P = pressure

V = volume

n = number

T = temperature

(and R is just a number)



T \uparrow : they go faster

P: is caused by them hitting the walls

Arrhenius Kinetik

<https://www.khanacademy.org/science/chemistry/acids-and-bases-topic/acids-and-bases/a/arrhenius-acids-and-bases>

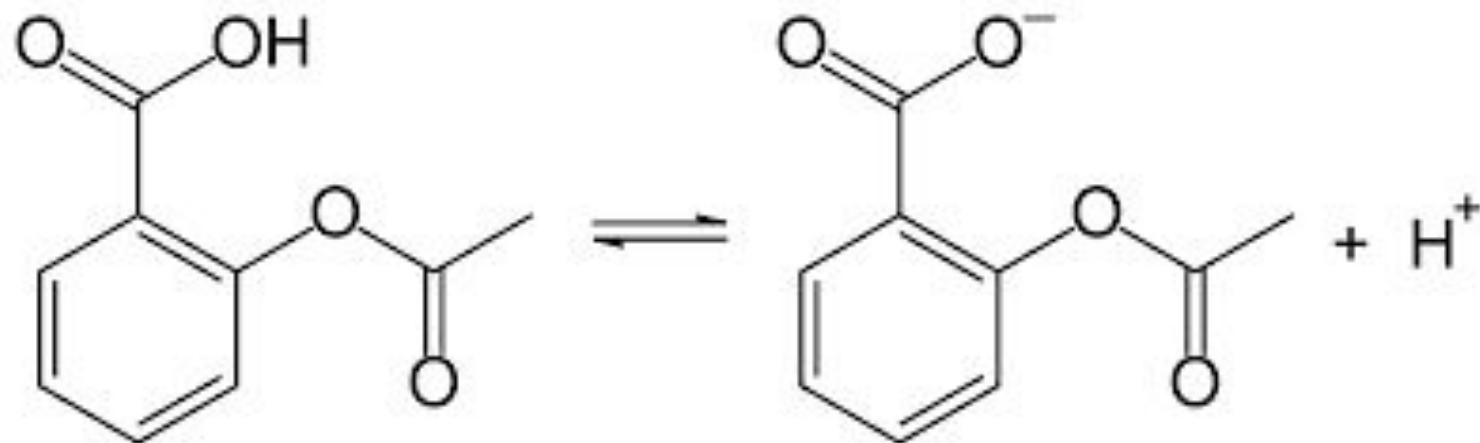
Arrhenius

By 1890 it was common knowledge that higher temperatures speed up reactions, often doubling the rate for a 10-degree rise, but the reasons for this were not clear.

Finally, in 1899, the Swedish chemist Svante Arrhenius (1859-1927) combined the concepts of activation energy and the Boltzmann distribution law into one of the most important relationships in physical chemistry



Arrhenius equation



$$(K_a = 2.75 \times 10^{-9})$$

Arrhenius

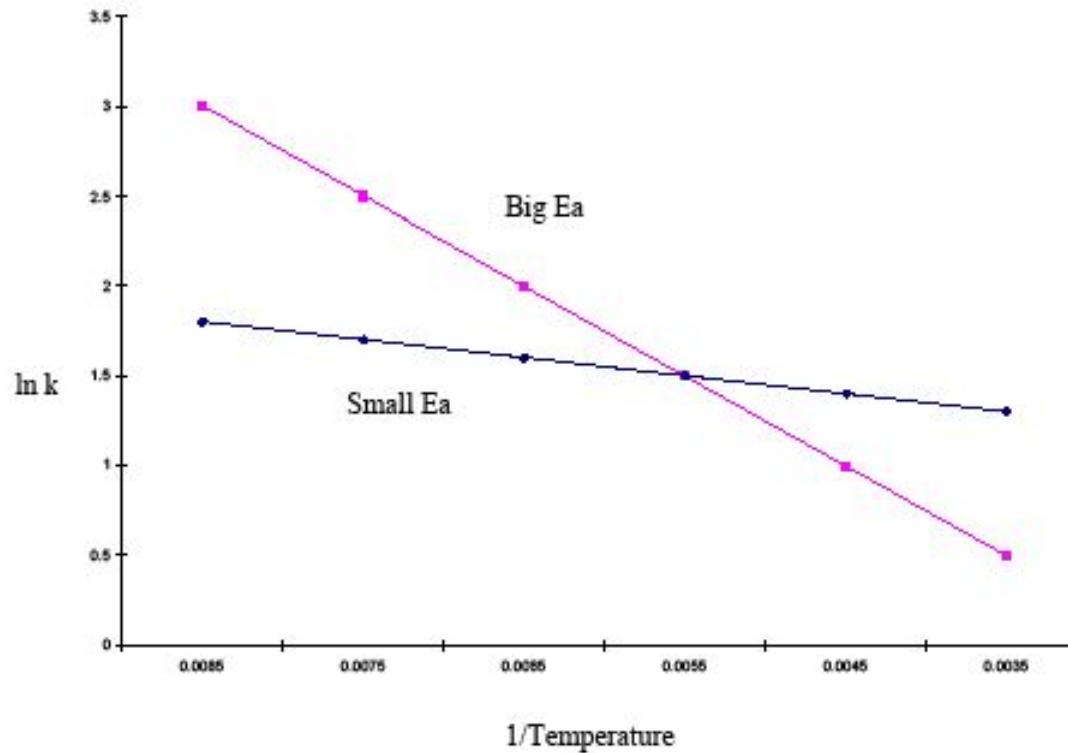
$$k = A e^{-\frac{E_a}{RT}}$$

Diagram illustrating the Arrhenius equation components:

- pre-exponential factor (A)
- activation energy (E_a)
- average kinetic energy ($\frac{E_a}{RT}$)

The diagram shows the Arrhenius equation $k = A e^{-\frac{E_a}{RT}}$. The term A is labeled "pre-exponential factor". The term E_a is labeled "activation energy". The term $\frac{E_a}{RT}$ is labeled "average kinetic energy". Arrows point from the labels to their respective terms in the equation.

Arrhenius



Material Science, MM/MD

(Cli WiSe24/12-19.01)

Material Science

Material Informatics

Material Informatics

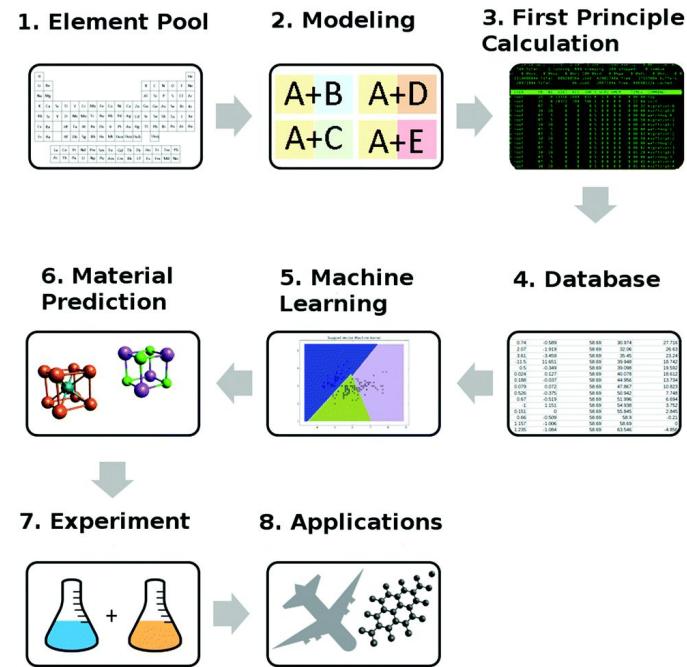
Materials informatics is a field of study that applies the principles of informatics and data science to materials science and engineering to improve the understanding, use, selection, development, and discovery of materials.

The term "materials informatics" is frequently used interchangeably with "data science", "machine learning", and "artificial intelligence" by the community. This is an emerging field, with a goal to achieve high-speed and robust acquisition, management, analysis, and dissemination of diverse materials data with the goal of greatly reducing the time and risk required to develop, produce, and deploy new materials, which generally takes longer than 20 years. [1][2][3] This field of endeavor is not limited to some traditional understandings of the relationship between materials and information.

Some more narrow interpretations include combinatorial chemistry, process modeling, materials databases, materials data management, and product life cycle management. Materials informatics is at the convergence of these concepts, but also transcends them and has the potential to achieve greater insights and deeper understanding by applying lessons learned from data gathered on one type of material to others. By gathering appropriate meta data, the value of each individual data point can be greatly expanded.

Takahashi, Keisuke, and Yuzuru Tanaka. 2016. "Materials Informatics: A Journey towards Material Design and Synthesis." *Dalton Transactions*. The Royal Society of Chemistry. June 6.

<https://pubs.rsc.org/en/content/articlelanding/2016/dt/c6dt01501h/unauth>.



"Materials Informatics." 2023. Wikipedia. Wikimedia Foundation. December 3. https://en.wikipedia.org/wiki/Materials_informatics.

MM, MD -
Molecular
Mechanics and
Dynamics

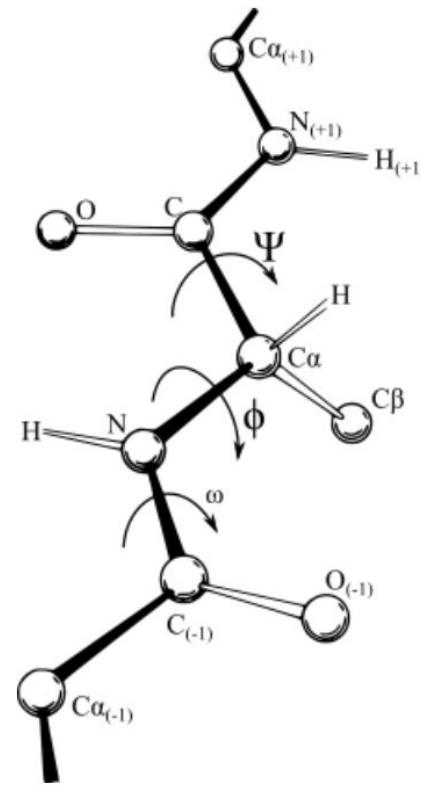
Molecular Mechanics (MM)

Molecular Mechanics (MM)

Molecular mechanics is one aspect of molecular modelling

Involves the use of classical mechanics (Newtonian mechanics) to describe the physical basis behind the models.

Molecular models typically describe atoms (nucleus and electrons collectively) as point charges with an associated mass.

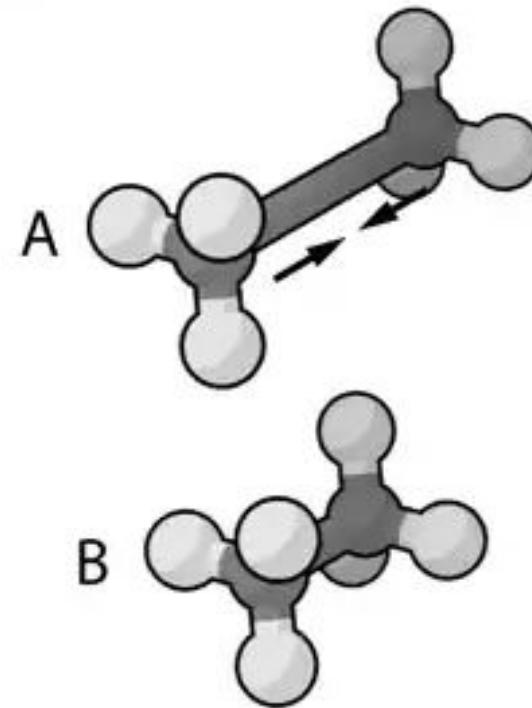


Molecular model of a protein

Force Fields

Force Field

Force fields are interatomic potentials and utilize the same concept as force fields in classical physics, with the difference that the force field parameters in chemistry describe the energy landscape, from which the acting forces on every particle are derived as a gradient of the potential energy with respect to the particle coordinates.



A force field is used to minimize the bond stretching energy of this ethane molecule.

Example: openMM - Force Fields

When you create a force field, you specify one or more XML files from which to load the force field definition. Most often, there will be one file to define the main force field, and possibly a second file to define the water model (either implicit or explicit). For example:

```
forcefield = ForceField('amber14-all.xml', 'amber14/tip3pfb.xml')
```

Modeling HW

Example: openMM/CUDA

When creating a **Simulation**, you can optionally tell it what **Platform** to use. OpenMM includes four platforms: **Reference**, **CPU**, **CUDA**, and **OpenCL**. The following lines specify to use the **CUDA** platform:

```
platform = Platform.getPlatformByName('CUDA')

simulation = Simulation(prmtop.topology, system, integrator, platform)
```

The platform name should be one of **OpenCL**, **CUDA**, **CPU**, or **Reference**. You also can specify platform-specific properties that customize how calculations should be done. See Chapter 11 for details of the properties that each Platform supports. For example, the following lines specify to parallelize work across two different GPUs (CUDA devices 0 and 1), doing all computations in double precision:

```
platform = Platform.getPlatformByName('CUDA')

properties = {'DeviceIndex': '0,1', 'Precision': 'double'}

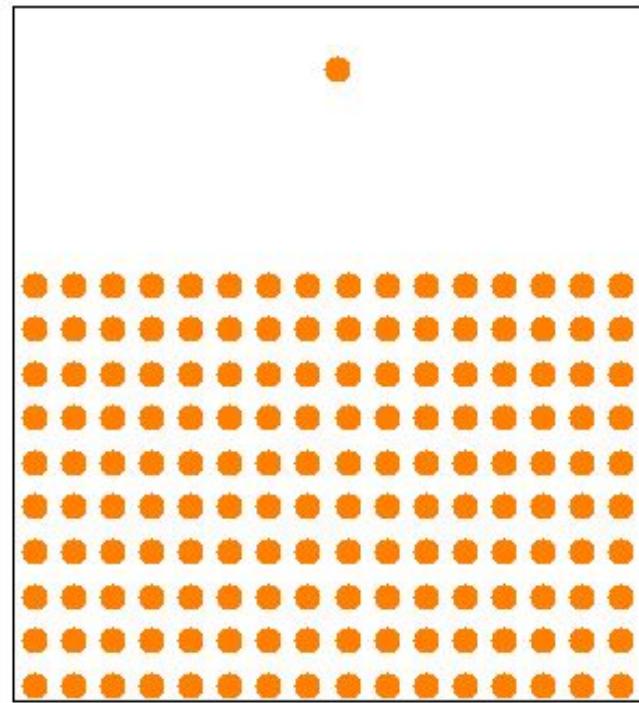
simulation = Simulation(prmtop.topology, system, integrator, platform, properties)
```

Molecular Dynamics

Molecular dynamics (MD)

Vast number of particles => impossible to determine the properties => using numerical methods

MD simulations are mathematically ill-conditioned, generating cumulative errors in numerical integration that can be minimized with proper selection of algorithms and parameters, but not eliminated.



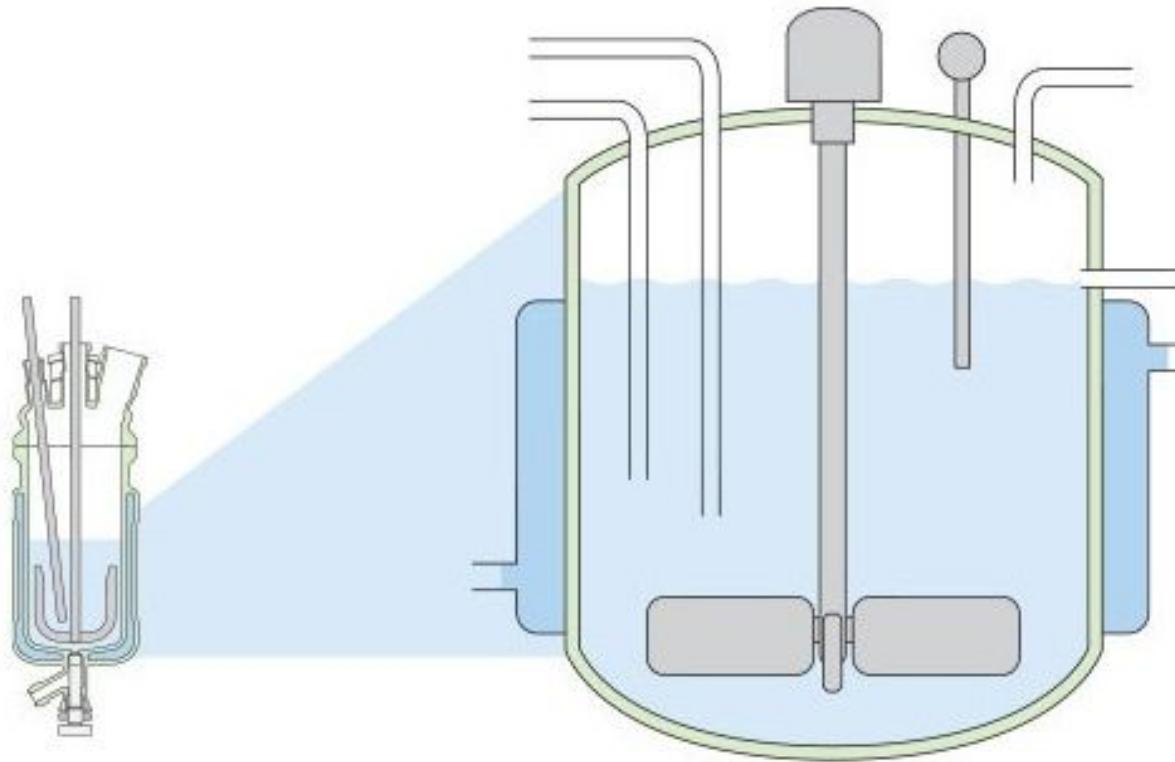
Deposition of one copper (Cu) atom on a cold crystal of copper. The kinetic energy of the atom approaching from the top is redistributed among the other atoms, so instead of bouncing off it remains attached due to attractive forces between the atoms.

Scale-Up

#NEW

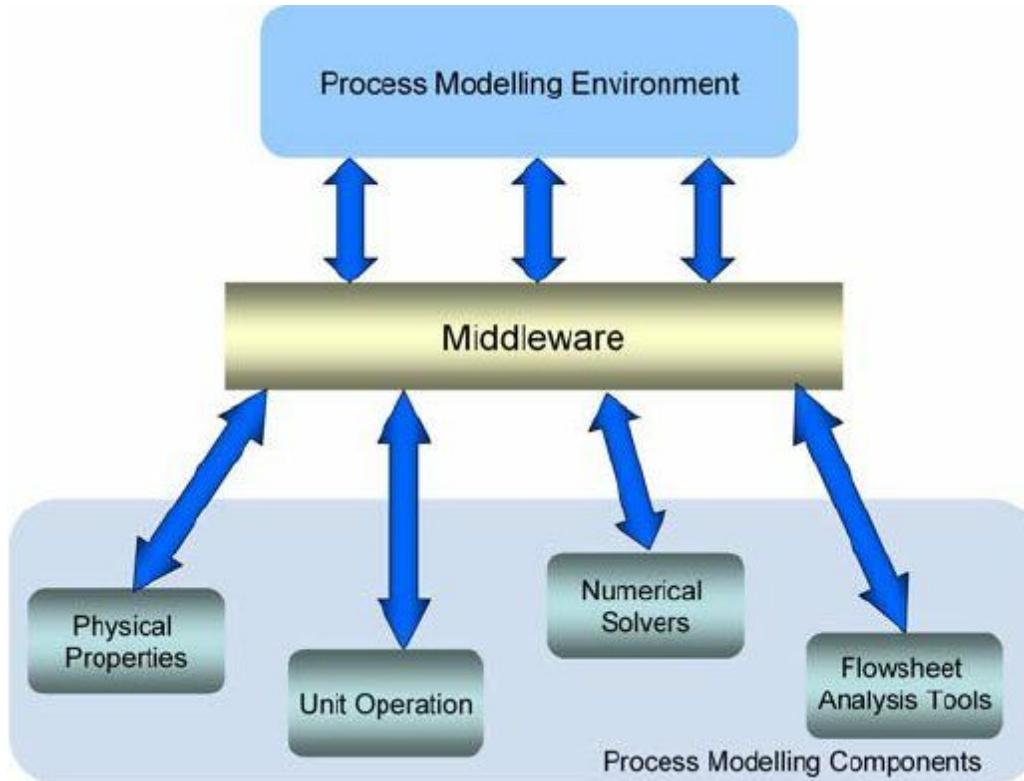
#essential

Scale-up



[https://www.creative-biolabs.com/drug-discovery/therapeutics/
process-and-scale-up.htm](https://www.creative-biolabs.com/drug-discovery/therapeutics/process-and-scale-up.htm)

Chemical process simulators

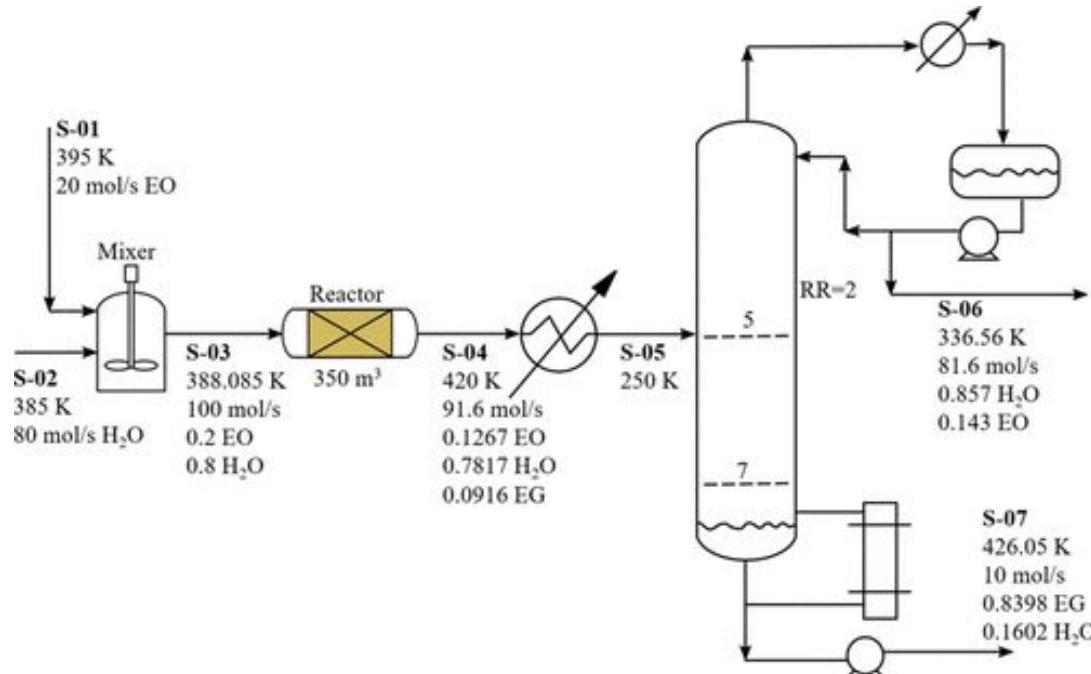


Ricardo Morales-Rodriguez

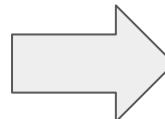
https://www.researchgate.net/publication/221910196_Interoperability_between_Modelling_Tools_MoT_with_Thermodynamic_Property_Prediction_Packages_Simulis_Thermodynamics_and_Process_Simulators_ProSimPlus_via_CAPE-OPEN_Standards/figures?lo=1

Modelica

Chemical Process Simulation Using OpenModelica



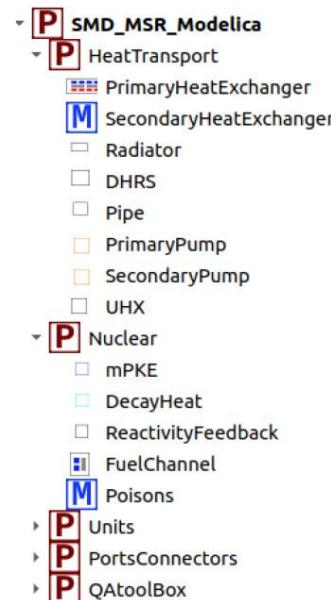
<https://github.com/FOSSEE/OMChemSim>



<https://pubs.acs.org/doi/10.1021/acs.iecr.9b00104> - Chemical Process Simulation Using OpenModelica

Modelica Unit operations

- Unit Operations
- Mixer
- Heater
- Heat Exchanger
- Cooler
- Valve
- Shortcut Column(FUG method)
- Compound separator
- Flash
- Splitter
- Centrifugal Pump
- Adiabatic Compressor
- Adiabatic Expander
- Conversion Reactor
- PFR
- Rigorous Distillation Column
- Rigorous Absorption Column



<https://www.sciencedirect.com/science/article/abs/pii/S0029549323003333>

```

model Condensor
  extends Simulator.UnitOperations.DistillationColumn.Cнд;
  extends Simulator.Files.ThermodynamicPackages.NRTL;
end Condensor;

model Discolumn extends Simulator.UnitOperations.DistillationColumn.DistCol;
Condensor condenser(Nc = Nc, C = C, CType = CType, Bin = Bin_t[1]);
Reboiler reboiler(Nc = Nc, C = C, Bin = Bin_t[Nt]);
Tray tray[Nt - 2](each Nc = Nc, each C = C, Bin = Bin_t[2:Nt - 1]);
end Discolumn;

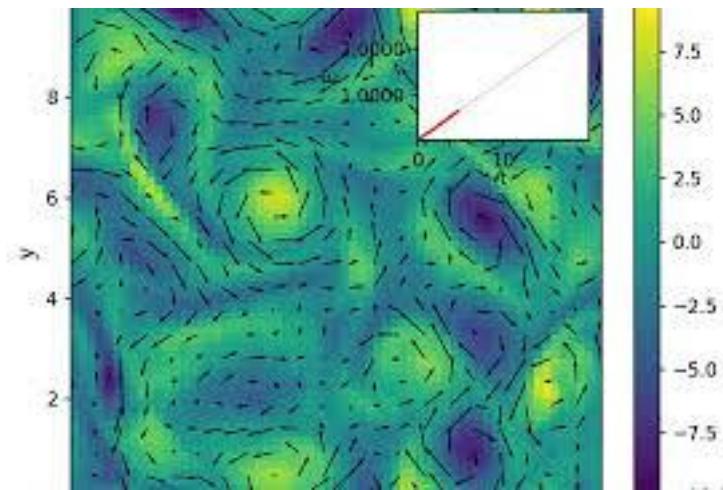
model flowsheet
  import data = Simulator.Files.ChemsepDatabase;
  //instantiation of Ethanol parameter data.Ethanol eth;
  //instantiation of Toluene parameter data.Toluene tol;
  //Number of Components
  parameter Integer Nc = 2;
  parameter Simulator.Files.ChemsepDatabase.GeneralProperties C[Nc]={eth,tol};
  // definatation of the unit operations and streams used
  PSD.Discolumn dis1(Nc = Nc, C = C, Nt = 35, Ni = 2, InT_s = {6, 16}));
  PSD.matstream feed(Nc = Nc, C = C);
...

```

CFD

Computational Fluid Dynamics

- [https://lorenabarba.com/blog/cfd-pyton-12-steps-to-navier-stokes/](https://lorenabarba.com/blog/cfd-python-12-steps-to-navier-stokes/)
- <https://fluidsim.readthedocs.io/en/latest/> Fluidsim is a framework for studying fluid dynamics with numerical simulations using Python. It is part of the wider project FluidDyn.
<https://fluidsim.readthedocs.io/en/latest/examples/running-simul-onlineplot.html>



CompChem, QM/QC

(Cli WiSe24/14-02.02-x)

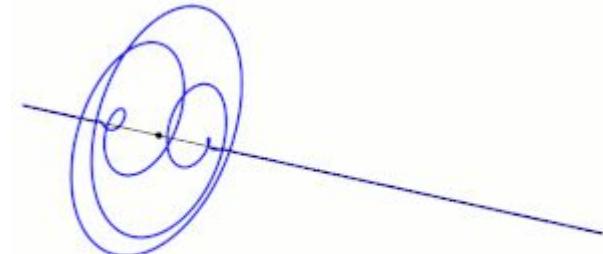
Schrödinger Equation („Wave mechanics“)

The Schrödinger equation is a linear partial differential equation that governs the wave function of a quantum-mechanical system.

Its discovery was a significant landmark in the development of quantum mechanics. The equation is named after Erwin Schrödinger, who postulated the equation in 1925 and published it in 1926, forming the basis for the work that resulted in his Nobel Prize in Physics in 1933.

Time-independent Schrödinger equation (general)

$$\hat{H} |\Psi\rangle = E|\Psi\rangle$$



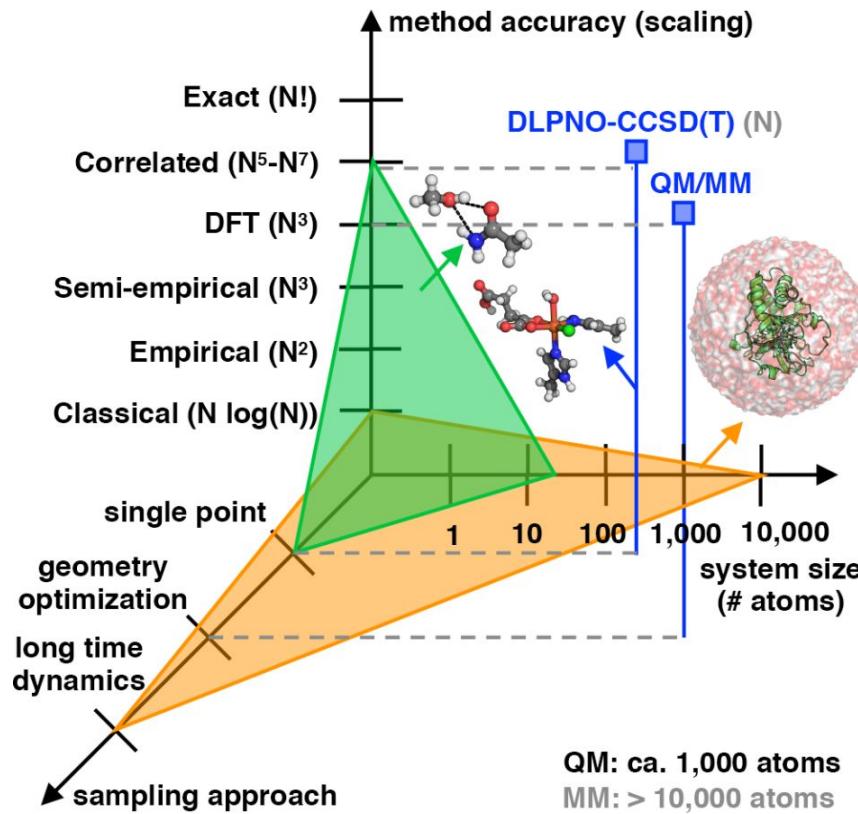
QCSoftware

(classic)

Cheminformatics	Free Software	Avalon Cheminformatics Toolkit · Bioclipse · Blue Obelisk · Chemistry Development Kit · ECCE · JOELib · OELib · Open Babel · RDKit
	Proprietary	Canvas · Chemicalize · Discovery Studio
Chemical kinetics	Free Software	APBS · Cantera · KPP
	Proprietary	Autochem · Chemical WorkBench · CHEMKIN · COSILAB · DelPhi · Khimera
Molecular modelling and visualization	List of molecular graphics systems	
	Free Software	Ascalaph Designer · Avogadro · BALL · Bikit · Gabedit · Ghemical · Jmol · Molekel · PyMOL · QuteMol · RasMol
	Proprietary	Abalone · ACD/ChemSketch · Atomistix ToolKit · ChemDraw · ChemWindow · EzMol · Gaussian · Maestro · MarvinSketch · MarvinView · MODELLER · Molecular Operating Environment · SAMSON · Spartan · UCSF Chimera · VMD
Molecular docking	List of protein-ligand docking software	
	Free Software	AutoDock · AutoDock Vina · FlexAID · rDock
	Proprietary	Glide · LeDock · Molecular Operating Environment
Molecular dynamics	Free Software	CP2K · GROMACS · LAMMPS · OpenMM · PLUMED
	Proprietary	Abalone · AMBER · CHARMM · CPMD · Desmond · GROMOS · NAMD
Quantum chemistry	List of quantum chemistry and solid-state physics software	
	Free Software	ABINIT · ACES (CFOUR) · AIMAll · BigDFT · CONQUEST · CP2K · Dalton · DP code · FLEUR · FreeON · MADNESS · MPQC · NWChem · Octopus · ORCA · PARSEC · PSI · PyQuante · PySCF · Quantum ESPRESSO (PWscf) · RMG · SIESTA · VB2000 · YAMBO code
	Proprietary	ADF · AMPAC · DMol3 · CADPAC · CASINO · CASTEP · COLUMBUS · CPMD · CRUNCH · CRYSTAL · DIRAC · Firefly · GAMESS (UK) · GAMESS (US) · Gaussian · Jaguar · MOLCAS · MOLPRO · MOPAC · ONETEP · OpenAtom · ORCA · PLATO · PQS · Q-Chem · Quantemol · Scigress · Spartan · TeraChem · TURBOMOLE · VASP · WIEN2k · XMVB
Skeletal structure drawing	Free Software	BKChem · JChemPaint · Molsketch · XDrawChem
	Proprietary	ACD/ChemSketch · BIOVIA Draw · ChemDoodle · ChemDraw · ChemWindow · JME Molecule Editor · MarvinSketch
Others	Aqion · Eulim · EXC code · GenX · GSim · Mercury · CrystalExplorer · ICM (ICM-Browser) · Materials Studio · Molden · OpenChrom · SASHIMI	

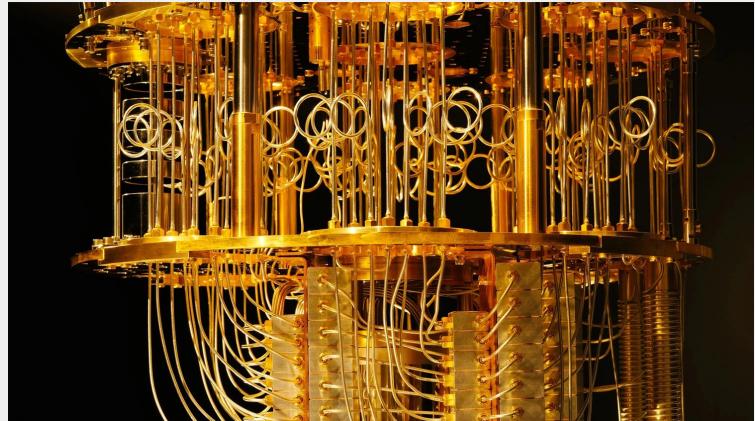
QM Runtime considerations

QM Runtime considerations



Vennelakanti, Vyshnavi, Azadeh Nazemi, Rimsha Mehmood, Adam H. Steeves, and Heather J. Kulik. 2022.
 "Harder, Better, Faster, Stronger: Large-Scale QM And QM/MM For Predictive Modeling In Enzymes And Proteins". Current Opinion In Structural Biology 72: 9-17. doi:10.1016/j.sbi.2021.07.004.

Quantum Chemistry and Quantum Computing



General procedure

Performing quantum chemistry + calculations on quantum computers → means using *algorithms* that have *supremacy* on quantum computers (which means they are hard to calculate on classic computers).

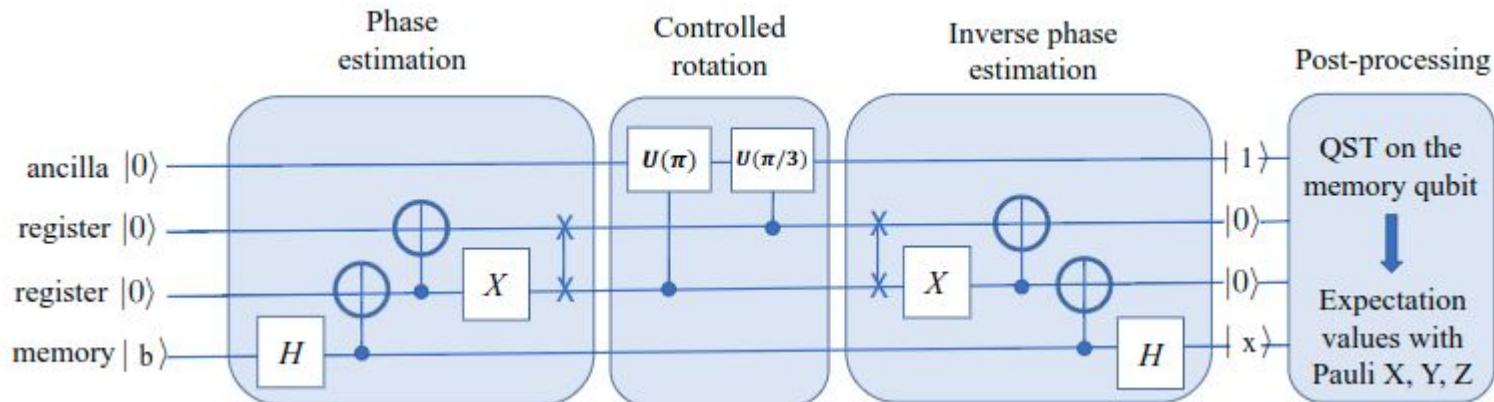
All current *quantum chemical calculations* on quantum computers → use standard QM software that has special plugins → to use these quantum algorithms → that run best on quantum processors.

Q-concepts — Q-computing — Q-Algorithms ┌ Q-Chemistry on Q-computers
QChemistry software └

Qiskit



- <https://qiskit.org/documentation/nature/> -
https://qiskit.org/documentation/nature/tutorials/10_lattice_models.html



<https://arxiv.org/abs/1804.03719>

