Integrated Data-driven materials science and digitalisation

Week 2 - Lecture 3

Professor Adham Hashibon

UCL - Institute for Materials Discovery - IMD

1. October 2025

Materials Informatics: The path for Materials Discovery

- We are in the data era!
- We are surrounded with massive amounts of data resulting from computer technologies, scientific tools, the Internet of things, etc.,
- **How does this impact materials science?**

Materials Informatics!

- *Materials informatics* is the application of **data science** to problems in **materials** science
- Some resources:


J. M. Rickman, T. Lookman, and S. V. Kalinin, "Materials Informatics: From the Atomic-Level to the Continuum," *Acta Materialia* 168 (April 2019): 473–510. https://doi.org/10.1016/j.actamat.2019.01.051


L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, "Data-Driven Materials Science: Status, Challenges, and Perspectives," *Advanced Science* 6, no. 21 (2019): 1900808. https://doi.org/10.1002/advs.201900808


Xiao-Qi Han, Xin-De Wang, Meng-Yuan Xu, Zhen Feng, Bo-Wen Yao, Peng-Jie Guo, Ze-Feng Gao, Zhong-Yi Lu, "AI-driven inverse design of materials: Past, present and future," arXiv preprint (2024). https://arxiv.org/abs/2411.09429


A. Agrawal and A. Choudhary, "Deep materials informatics: Applications of deep learning," *MRS Communications* (2019). https://link.springer.com/article/10.1557/mrc.2019.73

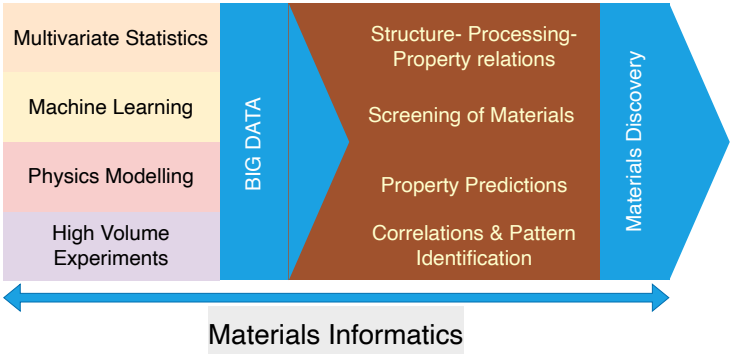
But wait a minute. What is Data Science actually ?

- It is the science of data or the study of data
- It is a multidisciplinary field comprising:
  - statistics, and analytics, visualisation,
  - data management, data mining, databases,
  - informatics and computer science,
  - knowledge representation and ontology
  - artificial intelligence, machine learning,...

But wait another minute... What is *informatics* actually ?

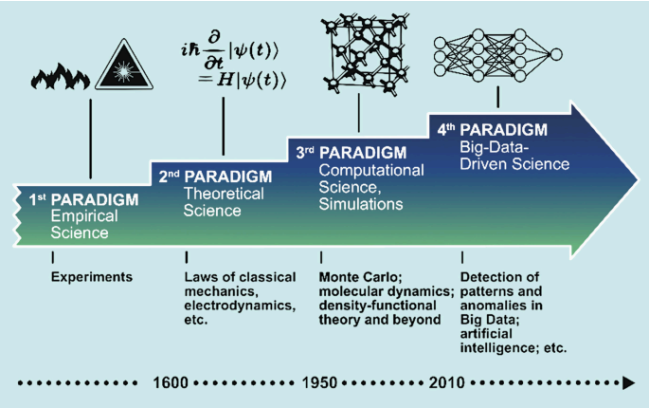
- Data acquisition from experiments, sensors, databases
- Data curation (cleaning, metadata, ontology, standardisation)
- Representation (defining features, e.g. descriptors, fingerprints)
- Analysis, statistical or machine learning models
- Inference, prediction and discovering patterns or predicting outcomes
- Knowledge integration, and linking results back to theory or design

Materials informatics is the application of data science to problems in materials science



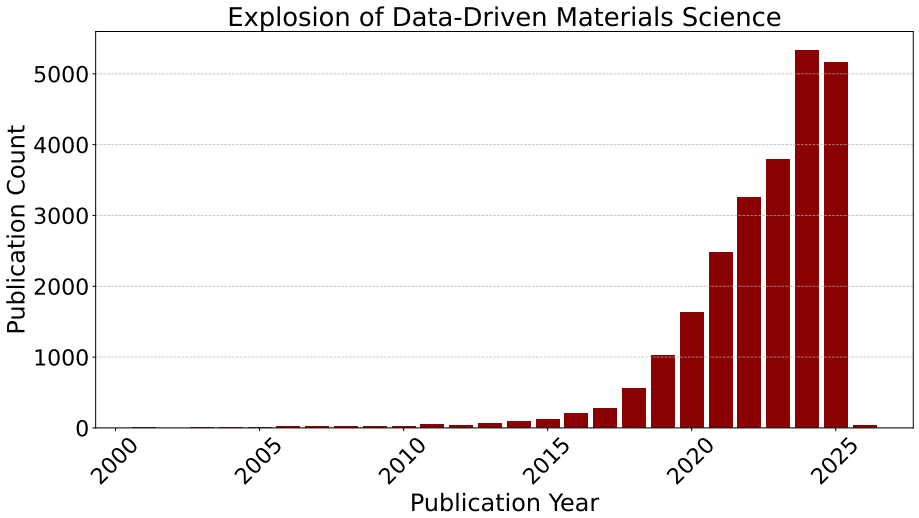
Materials Informatics combines various existing approaches into the integrated data driven approach!

A new way to do materials discovery - The Data Driven Paradigm

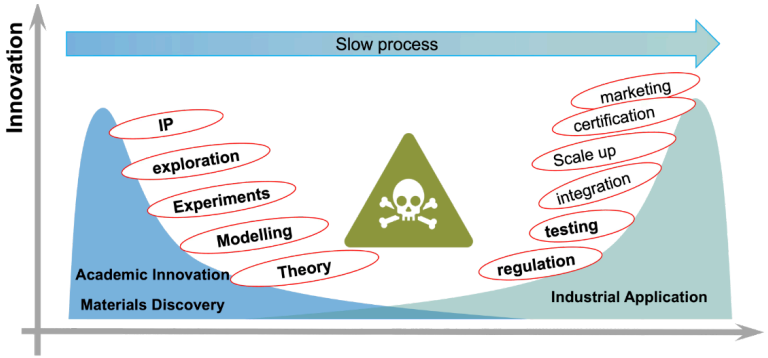


https://www.google.com/url?sa=i&url=https%3A%2F%2Flink.springer.com%2Frwe%2F10.1007%2F978-3-319-44677-6_104&psig=AOvVaw0AhVZ2rnF1FJqoOAbOvBpM&ust=1759090327567000&source=images&cd=vfe&opi=89978449&ved=0CBkQjhxqFwoTCIDf-aPLkZADFQAAAAAdAAAAABAE

Publication Growth in ML for Materials Discovery

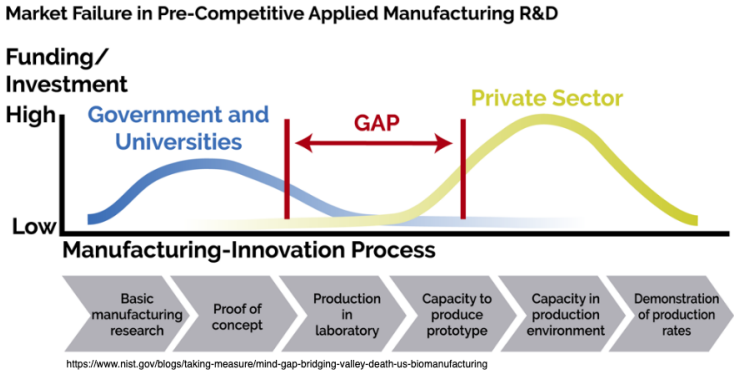


Breaking barriers to materials discovery



The Innovation "Valley of Death"

Breaking barriers to materials discovery

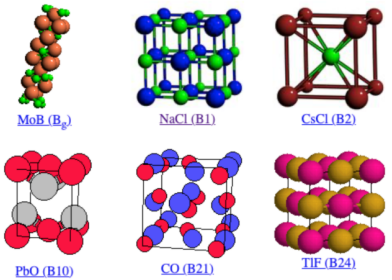


The Innovation "Valley of Death"

Historical Perspective: Is Data-Driven a New Thing?

Structureal Classification of binary AB Solids

- Aim to understand bonding and discover new materials (more than 50 years ago!)
- Crystal structure classification of of AB binary solids into rocksalt, zincblende, wurtzite, cesium chloride, and diamond structures
- Octet rule: $N_A + N_B = 8$: the total number of valence electrons, or $A^N B^{8-N}$.



Prototypes

Table: Common AB-type binary crystal structures shown in the figure.

Prototype	StrukB	Lattice Type	Examples
MoB	B9	Orthorhombic	Transition metal borides
NaCl	B1	Face-centered cubic	NaCl, MgO, TiN
CsCl	B2	Simple cubic	CsCl, β-brass (CuZn)
PbO	B10	Tetragonal	PbO, SnO
CO	B21	Orthorhombic	CO, FeSi
TiF	B24	Orthorhombic	TiF, MnF

Early use of Features/Descriptors

PHYSICAL REVIEW B VOLUME 17, NUMBER 6 15 MARCH 1978

Quantum-defect theory of heats of formation and structural transition energies of liquid and solid simple metal alloys and compounds

J. R. Chelikowsky and J. C. Phillips
Bell Laboratories, Murray Hill, New Jersey 07974
(Received 9 November 1977)

This situation can be formulated quantitatively as follows. Structural energy differences are, for the most part, too small to be calculated quantum mechanically, because such calculations require self-consistent crystal potentials of an accuracy beyond the present state of the art. However, if we consider the problem from the point of view of information theory, then the available structural data already contain a great deal of information: about 120 bits, in the case of the $A^N B^{8-N}$ octet compounds. Thus one can reverse the problem, and attempt to extract from the available data quantitative rules for chemical bonding in solids.

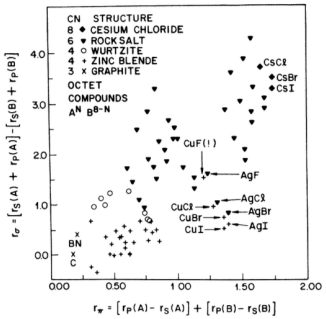


FIG. 9. St. John–Bloch plot for 79 binary octet crystals $A^N B^{8-N}$. Note in particular the separation of the wurtzite from the zinc-blende structures. This separation was not possible with the dielectric method. CuF, although tabulated by Wyckoff (Ref. 28) as a zinc-blende binary, does not exist as a stable compound (see text).

Good Descriptors

- DFT energies are more accurate measures of stability, but can discontinuously across systems for our choices of structure and boundaries, etc.
- Descriptors are chosen such they vary smoothly and monotonically across chemical space

Clustering in Descriptor Space (Pettifor 1984)

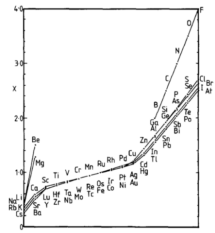


Fig. 1. The chemical scale χ.

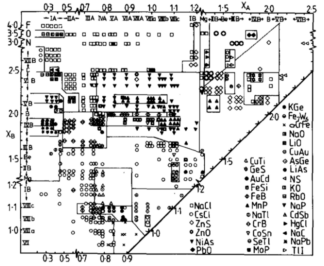


Fig. 2. The (χ_A, χ_B) structure map for 574 AB compounds.

χ is set up by requiring no overlap or mixing of neighbouring groups and constrained to vary linearly across the transition-metal series

Ashby Plot

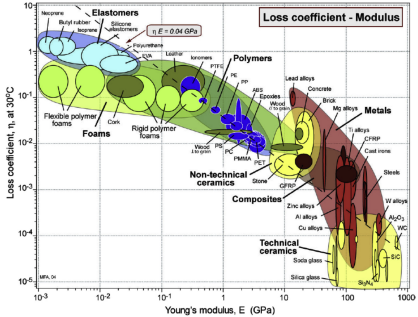
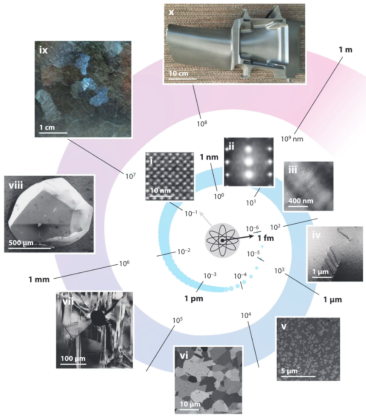


Fig. 3. Ashby plot of the loss coefficient versus Young's modulus for different classes of materials. The loss coefficient measures the internal friction or damping, an important material property when structures vibrate or dissipate energy. In metals, a large part of the loss is hysteretic, caused by dislocation movement; it is high in soft metals such as lead and aluminum. Alloyed metals like bronze and high-carbon steels have relatively low loss because solute atoms pin dislocations. Exceptionally high loss is found in the Mn-Cu alloys because of a strain-induced martensitic transformation. In polymers, chain segments slide against each other when loaded; their relative motion dissipates energy. (reproduced from Ref. [19]).

Materials are Complex Hierarchical Systems. . .

Length scales from atomistic to macroscopic, starting at the center and moving clockwise in increasing microstructural length scale:

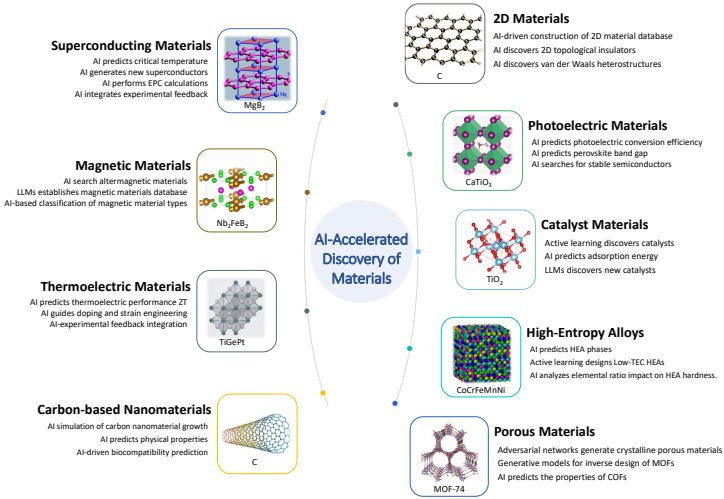
- Atomic-resolution image of $BaTiO_3$ along [001]
- Diffuse electron scattering in [112] zone axis orientation, indicating the presence of short-range order in Cu–15%Al
- Lamellar poly(styrene-*b*-isoprene) block copolymer microstructure
- Dislocation array in Cu–Al
- Dendritic γ precipitates in a Rene-88DT superalloy
- Polycrystalline grain microstructure in an IN100 superalloy
- Type-II twins and magnetic contrast in a multiferroic Ni_2MnGa alloy
- A centimeter-size single extracted grain of Ni_2MnGa
- Centimeter-sized grains in architectural titanium
- Single-crystal superalloy turbine blade (courtesy of T.)



Dive into materials!

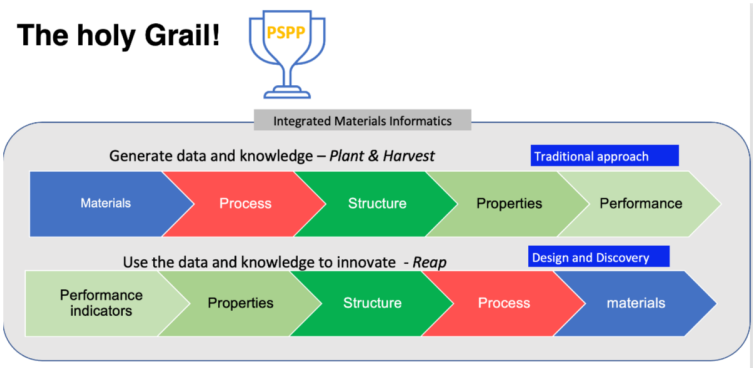
Watch the video: **Dive in (YouTube)**

AI-driven discovery of materials

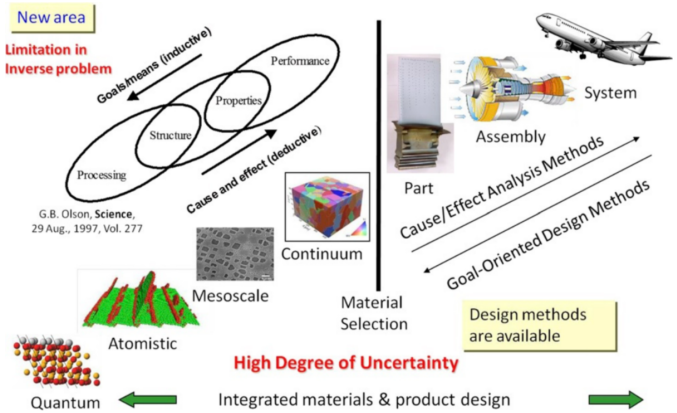


Han, Xiao-Qi, et al. "Ai-driven inverse design of materials: Past, present, and future." Chinese Physics Letters 42.2 (2025): 027403.

Inverse Design



Data Enables Inverse, Targeted Design



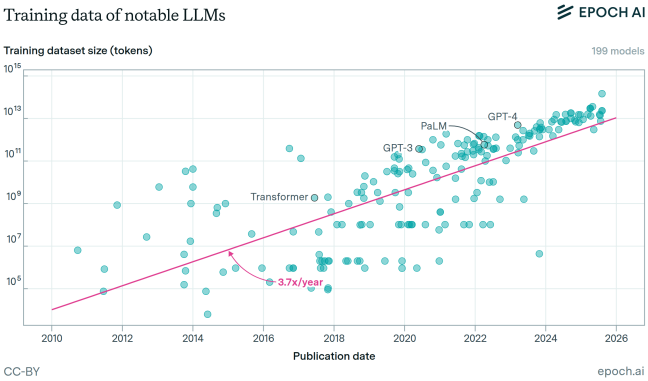
What are the main challenges?

- In materials we are fighting not only model complexity but also extremely high dimensional design space
- The materials design space as we will see is composed of $10's$ to $1000's$ to ... *millions* of parameters!
  - composition (all the possible combinations in the periodic table!)
  - structure (how many crystallographic systems and space groups are there?)
  - processing variables (sintering, quenching, calendaring, annealing, mixing, rolling, milling, deposition ....)

The data Challenge: Small Versus Large Data Sets

- Relatively small, **heterogeneous** data sets
- Existing data is often confined to specific regions of parameter (e.g., composition) or for a narrow range of systems.
- We aim is to extract maximum information from limited data (small data sets $\rightarrow$ overfitting, poor generalization)
- however good studies need $\sim 1000's$ training samples

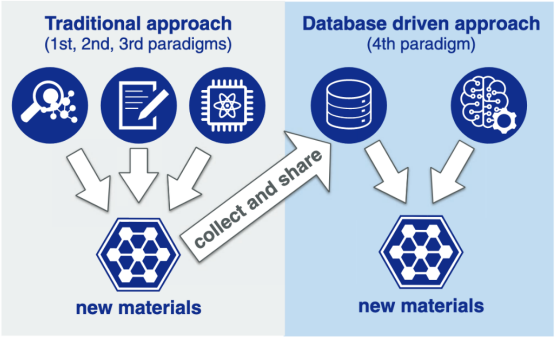
Machine learning models are very data intensive!



Source: Robi Rahman and David Owen (2024), "The size of datasets used to train language models doubles approximately every six months". [source: epoch.ai]

Materials Informatics is *very* data greedy!

So we need lots and lots of data!



Himanen et al., "Data-Driven Materials Science.

Some Materials Data Infrastructures

Name	Website	Short Description
nanoHUB	nanohub.org	Online simulation, data sharing, and education platform for nanoscience
Materials Project	materialsproject.org	Computed materials properties and structures (DFT)
Materials Cloud	materialscloud.org	High-throughput DFT via AiiDA
AFLOW	aflowlib.org	Automated framework for materials discovery and DFT data
QCArchive	qcarchive.molssi.org	Quantum chemistry, molecular simulation aggregator
COD	crystallography.net	Crystallographic structure database (IUCr community)
Materials Data Facility	materialsdatafacility.org	Cloud-based storage and access infrastructure for materials data
NOMAD CoE	nomad-coe.eu	Repository for computational materials and metadata provenance
OQMD	oqmd.org	Open Quantum Materials Database (computed properties)
Open Materials DB	openmaterialsdb.se	Open repository integrating multiple data sources

Introduction to the DiscoMat Platform and Linux

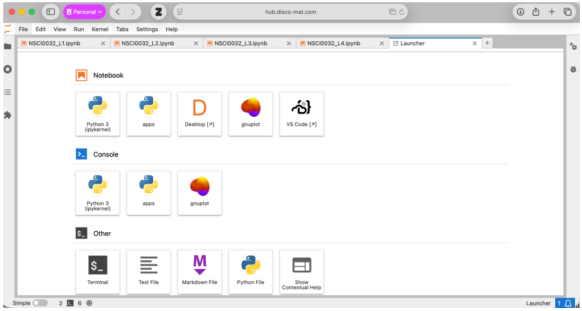
Based on Ubuntu's "Command Line for Beginners"

Professor Adham Hashibon

UCL - Institute for Materials Discovery - IMD

9.10.2025

The Materials Discovery Platform - I



- A multi-user, cloud-based, self-hosted open-source project developed at IMD by the group of Professor Hashibon.
- Integrates and extends several state-of-the-art scientific and data science components.
- Based on a JupyterHub Technology

The Materials Discovery Platform- II

- Design principles:
  - Containerisations and virtualisation
  - Cloud based
  - Self hosted, data is within UCL
  - Secure and behind fire wall
  - Linux (Unix) based
  - Support Desktop (GUI: Graphical User Interface) and Command Line
  - Powerful, High Performance Computing tools (including GPU) under the hood.
- Benefits to Users
  - Everything ready out of the box!
  - centrally managed
  - Requires a web browser only

The Materials Discovery Platform- III

- Features/Components
  - Terminal Emulator (no GUI)
  - Bash (and other) Unix Shells
  - Jupyter Notebook
  - CONDA pre set with mamba for fast install
  - Full stack Data Science Kernel (`conda activate /apps/conda`) including Keras, SciKit Learn, Tensor Flow, numpy/scipy, blas, pandas, matplot lib, and more
  - Simulation tools like Lammmps and QuntumEspresso
  - Online Vscode server
  - Fully fledged Desktop Linux System!
  - Pre and post processing tools and parallel data and computational workflows

The Materials Discovery Platform- III

CPU Architecture

- Architecture: x86_64
- CPU(s): 128
- Model name: Intel(R) Xeon(R) Platinum 8358 CPU @ 2.60GHz

Memory Overview

- Total Memory: 503 GiB

The Materials Discovery Platform - IV

The User Interface

- Launcher
- Apps
- File Manager
- Terminal (shell)

The Linux Desktop

- GUI based
- File Manager (like Finder or Explorer in windows)
- Terminal Emulator (a GUI emulation of a terminal)

VSCODE Server

The Materials Discovery Platform - IV

The User Interface

- Launcher
- Apps
- File Manager
- Terminal (shell)
- Python notebooks
- Gnuplot notebook

The Linux Desktop

- GUI based
- File Manager (like Finder or Explorer in windows)
- Terminal Emulator (a GUI emulation of a terminal)

VSCODE Server

Activity 1

Explore the native components

- Launcher
- Apps
- File Manager
- Terminal (shell)
- Python notebooks
- Gnuplot notebook

Activity 2

Explore the "proxied" components

- Desktop $\longrightarrow$ open terminal
- Vscode server

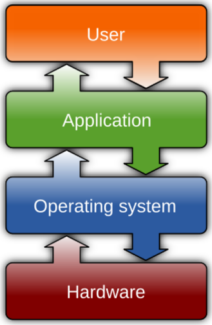
Activity 3

Explore JupyterHub framework

- Manage sessions
- Manage backend "virtual machines"

Operating System (OS)

- An operating system (OS) is system software that manages computer hardware and software resources, and provides common services for computer programs.



OS architecture as a bridge hardware and software to user

The Unix OS - I

- It is the most powerful operating system ever made!
- Invented and developed since 1969 by Ken Thompson at AT&T Bell Labs
- In 1991 a finish computer science student started a small project... today it is the new Unix, well LINUX...
- Today Linux/Unix it power the internet, it dominates the scientific and other server market!
- MacOS, iOS, iPadOS, Android are all UNIX based (with additional layers, especially GUI)

The Unix OS - II

- Its main principles:
  - minimalistic design - small and simple is beautiful and elegant
  - not designed to be easy to use, but to be **user-useful and super powerful for data and workflows!**
  - make each program do one thing well, and only one thing!
  - portability over efficiency
  - no complex user interfaces
- System abstraction
  - Kernel: hardware layer
  - Shell: text mode layer
  - X Windows: GUI layer

What is the Command Line?

- A text based *command line interface* to control your computer (communicate with the file system, memory, and other programmes)
- Enables command execution, scripting, and automation!
- Also called the terminal, console, or shell
- Common shells: `bash`, `zsh`, `fish`
- Common Terminals: GnomeTerminal, Xterm, Konsole, Terminal (MAC), iTerm (Mac)

Moving Around the Filesystem

- `pwd` print current directory.
- `cd /`
- `cd ..`
- texttttcd
- Absolute: start with `/`.
- Relative: based on current location.

Listing and Creating Files

- `ls`: list files and directories.
- `mkdir name`: make directory.
- `touch file.txt`: create an empty file.
- `ls -a`: show hidden files.

Combining and Redirecting Commands

- `command > file` redirect output of command into file (overwrite).
- `command >> file` append output.
- `cat file1 file2` concatenate or display files.
- `ls | wc -l`. pipe output of one command to another.

Working with Hidden Files

- Files beginning with `.` (i.e., dot) are hidden by default
- `ls -a`: shows *all* files, including hidden files.
- You can access them normally: `cd .config`

Feel lost? .. The Linux Directory Structure for the rescue!!

Directory	Contents & purpose
/	Root of the entire filesystem, everything starts here. Nothing above it.
/bin	Core binary commands, like shell ones, e.g. cp, mv, ls).
/usr/bin	Standard user programs and utilities
/usr/local/bin	Locally installed, site-specific tools and scripts.
/sbin	System binaries for administration by superuser or root.
/lib	Core system libraries
/usr/lib	Libraries supporting user tools
/opt	Optional or third-party software packages
/apps	Shared application space, platform wide installations.
/apps/conda	Centralized Conda/Mamba environment containing scientific and AI toolchains (e.g., Python, TensorFlow, QE, LAMMPS).
/tmp	Temporary storage for programs; writable by all users and typically cleared on reboot.
/home	Root of the users home, since there is one user in our case, its only a Jovyan folder there!
/etc	System wide configuration and initialisation files.
/dev	Device files representing hardware and virtual devices (e.g., disks, terminals, GPUs, cpus...).
/proc	Virtual filesystem exposing kernel and process information in real time (like cpuinfo)

Activity 4

Explore the File System

Explore the file system using the cd command, go all the way to the root, and down every folder, what do you find in it?

Activity 5

Lets create a simple automation

The above was a very gentle intro to the platform and command line.
Now we move up a notch, and really there is no need for us to repeat everything in slides, the internet is full of docs and user guides (see bibliography below for some).
And today, we are in the Chatbot era! so lets use it..

> *Use a Chatbot (preferred Copilot, the safe ccess for which is provided by UCL) to create an automated script that creates a series of folders names* `<some prefix>0001.run` *with a running index from 1 to 100, then in each folder creates a text file with the text* `lammps < in > out` *then waits 30 seconds, and goes back to each flolder and prints the content of the file.*

Sample additional resources! (in and outside UCL) I

Mitchell Anicas, *An Introduction to the Linux Terminal*, DigitalOcean, June 1, 2022. https://www.digitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal

Justin Ellingwood, *Basic Linux Navigation and File Management*, DigitalOcean, 2022. https://www.digitalocean.com/community/tutorials/basic-linux-navigation-and-file-management

Mitchell Anicas, *An Introduction to Linux Permissions*, DigitalOcean, June 3, 2022. https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-permissions

Command Line for Beginners, Ubuntu Tutorials. https://ubuntu.com/tutorials/command-line-for-beginners#1-overview

Course Catalogue — Advanced Research Computing, UCL, UCL ARC. https://www.ucl.ac.uk/advanced-research-computing/training/course-catalogue

Sample additional resources! (in and outside UCL) II

Moodle: Course 12953, UCL Moodle.
https://moodle.ucl.ac.uk/course/view.php?id=12953

An Introduction to Unix, Department of Computing, Imperial College London.
https://www.doc.ic.ac.uk/~wjk/UnixIntro/

Aristotle Platform How-To, Materials Discovery Group Tutorials, GitHub Repository.
https://github.com/materials-discovery/Tutorials/blob/main/aristotle_howto.md

File Management and Data Handling How-To, Materials Discovery Group Tutorials, GitHub Repository.
https://github.com/materials-discovery/Tutorials/blob/main/files_howto.md

Library Resources I

The Linux Command Line, 2nd Edition
By William E. Shotts
Publisher: No Starch Press (504 pages, February 2019)
A complete introduction to Bash and the Unix philosophy — from basic navigation and pattern matching to writing shell scripts and understanding how Linux works at a deeper level.

Efficient Linux at the Command Line
By Daniel J. Barrett
Publisher: O'Reilly Media (248 pages, February 2022)
Practical techniques for power users, sysadmins, and developers to build complex pipelines, automate workflows, and understand what happens behind the shell prompt.

Beginning the Linux Command Line, 2nd Edition
By Sander van Vugt
Publisher: Apress (416 pages, November 2015)
A task-oriented guide verified across major Linux distributions — ideal for those who prefer fast, keyboard-driven administration over GUI tools.

Library Resources II

Practical Linux Command Line
Video Course – O'Reilly Learning Platform
https://learning.oreilly.com/videos/practical-linux-command/10000DIVC2022145/