Data science: scraping, mining, exploratory data analysis strategies in life sciences

Bruno Miguel Soares Guerreiro, 58721

Programa Doutoral em Bioquímica (Especialidade em Biofísica)

Para obtenção da UC: Módulo Especializado em Biofísica II

Preamble

This report agglomerates the course certifications obtained in the field of data science, and the real-world application of the acquired knowledge into problem-solving the PhD thesis. Data science methodologies such as data scraping, data mining, exploratory data analysis, data visualization, machine learning and multidimensional analysis tools such as one-hot encoding and principal component analysis, were applied in various points of the doctoral thesis, namely in:

(i) data scraping algorithms for signal filtering in isochoric nucleation detection;

(ii) extremophile polysaccharide database (145x128 dimensions) generation and analysis;

(iii) contour plot visualizations to characterize FucoPol-based isochoric phase diagrams;

(iv) energy profile data visualization for reformulating Classical Nucleation Theory.

Course Certificates

Below you can find the official certifications to which the applied knowledge pertains:

- Data Science A–Z™, SuperDataScience (Udemy, 2018)
- Data Science Specialization, Johns Hopkins University (Coursera, 2020)
- Deep Learning Specialization, deeplearning.ai (Coursera, 2020)
- Deep Learning.AI TensorFlow Developer Specialization, deeplearning.ai (Coursera, 2020)
- Data Science Math Skills, Duke University (Coursera, 2020)
- Responsive Web Design Developer Certification, freeCodeCamp (freeCodeCamp.org, 2020)

Table of Contents

A brief goal-centric introduction to Data Science

In recent decades, the development of ever more complex high-throughput analytical methods has led to enormous amounts of collected data, which requires superhuman processing time and analytical abstraction to derive critical insights [1]. The datasets are often so complex, that simple relationships between variables can only provide a finite and minute amount of insight into elucidating intricate problems. These so-called hyperparameterized datasets are often multidimensional, meaning that comprehending all variable relationships often requires plotting thousands of linear relationships, performing hundreds of statistical testing and indulging in dozens of pre-processing algorithms to filter out less relevant data. The fundamental tenet in data science is *"data in, data out"*. In other words, the quality of the used data drives the quality of the insight and any data-driven decisions. For this reason alone, high-quality data science often requires the initial data to suffer several transformations before being visualized and discussed upon in scientific papers. Fields like bioinformatics and computational biology, which developed from the landscape of data science embedded into the life sciences, leverage the power of data analysis, machine learning and statistical modelling to derive valuable information. Depending on the goal, each subfield possesses their own "*cookbook*", which constitutes all data scraping, data mining and exploratory data analysis algorithms to transform datasets into verifiable knowledge, *e.g.* genomics and pharmacokinetics.

1.1. Modelling life science mechanisms for predictive analysis

Biological systems possess specialized mechanisms of action, which allow life to exist and perpetuate until a malfunction or cessation of said mechanisms occurs. These mechanisms possess a defined sequential pattern of functionality. Thus, they can be perceived as biological algorithms that sustain life. One of the great advantages of data science is not only the ability to integrate and analyze different data sources relative to experimental work done to probe and elucidate these biological mechanisms, but the ability to simulate *in silico* the outcome of said mechanisms. The whole field of pharmacokinetics for instance, relies on QSAR analysis (quantitative structure-function relationships) [2] to simulate the effects of a drug in the human body and effectively screen potential drug candidates before real-world application is even carried out. This not only provides a safe strategy towards minimizing biological drawbacks but allows to hasten the screening process several hundred times, which would otherwise be extremely sluggish at a laboratory scale and obstructive towards fast discovery.

1.2. Objective and general outlines

This report highlights the extensive usage of data science tools and techniques for rationalizing the multidimensional and multidisciplinary problems encountered herein. This doctoral thesis, which focuses on understanding the structure-function relationships of polysaccharides in cryopreservation, bears a unique similarity to the scientific rationale employed in QSAR analyses previously mentioned. Therefore, this report curates what can be considered a prototypical *cookbook* in polymeric cryopreservation research. Although the proper segmentation of a problem into individual segments to obtain linear relationships between predictor variables towards a specific function is one of the fundamental methodologies of the scientific method, some issues encountered in this thesis required a broader perspective. Understanding the acclimation of biological systems to cold environments has led to simultaneously studying ice binding and diffusion kinetics at the molecular scale, and multidimensional analysis of how cold-adapted microorganisms regulate their molecular consortium at a macroscale to survive extreme habitats. Both length scales have led to critical insights in order to design efficient cryoprotective applications and establish predictive rationales to what role polysaccharides may play in cryogenic survival.

For reading convenience, the author has decided to implement the description of fundamental data science concepts wherever they are used, instead of providing isolated definitions along the text. In this fashion, not only the concepts are better understood in their core, but an application-centric description of their utility allows to uncover the logical reasoning behind their usage. The following chapters will describe relevant applications where data science tools were deemed essential to solve a scientific problem, describing the issue encountered, the implementation of the algorithms and how they were built, using standard data science and programming terminology. The relevance of data science in this thesis has been instrumental.

Most scientific discovery presented herein has derived mostly from deeper analysis into correlations between data obtained from separate experiments, then by sequential experimental discovery of differential diagnosis designs in the laboratory. Therefore, the following sections highlight the major role that data science techniques had in deriving critical insights that are now published or in a review process and constitute the whole novelty of the PhD thesis to the collective body of knowledge in the literature.

`Application #1:` Data pre-processing algorithm for automated isochoric nucleation workflows

The isochoric nucleation detection (INDe) device [3], used in our paper "*Enhanced Control Over Ice Nucleation Stochasticity Using a Carbohydrate Polymer Cryoprotectant*", published in a collaboration with Prof. Boris Rubinsky from the University of California-Berkeley (USA), is a thermodynamic system that allows to detect the onset of ice nucleation by probing internal chamber pressure [4]. It allows for high-throughput data collection due to an automated freeze-thaw cycling and real-time monitoring of multiple nucleation temperatures as a time series. However, the raw data is obtained as a continuous function of pressure and temperature sensing over time, requiring manual determination of the nucleation temperature, which corresponds to a peak in internal chamber pressure (strain transduced as an electrical signal from the strain gauge). To automate the data collection phase, a personalized algorithm was built, which pre-processes the raw data, collects the nucleation temperature of each freeze-thaw cycle, performs mathematical modelling in real time and plots updated data visualizations (**Scheme 1**). This workflow was coded in Python, can be run in real-time with the INDe and provides paper-ready plots for scientific publishing. The public version of this algorithm was released under a CC BY-NC 4.0 license.

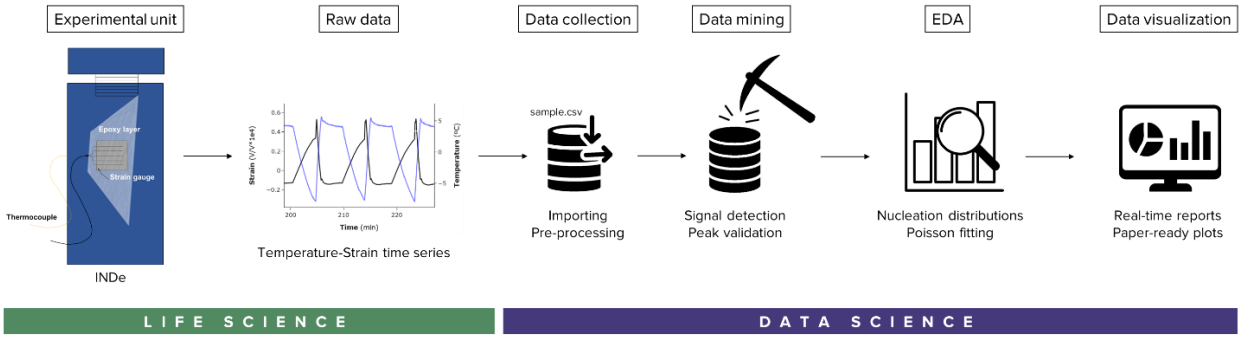


Figure 1. Data science workflow for INDe data collection, mining, exploratory analysis (EDA) and visualization.

2.1. Data import and processing

The interactive code for Stage 1 of the algorithm is available here (download and open).

After running at least N=300 freeze-thaw cycles from the isochoric chamber unit (INDe), the data is collected into an exportable *.csv* list with four variables: time, strain, set temperature and actual temperature (**Figure 2**). This arbitrary *sample.csv* file is used as data input into the data scraping algorithm after some numerical transformations in the time and strain domains.

```
In [17]: data.head()
```

Out[17]:

	Time	Strain	Tset	T	Temperature
0	0.000000	0.00000	5.0	5.1100	-5.1100
1	0.008367	-0.00221	5.0	5.1550	-5.1550
2	0.016700	-0.00087	5.0	5.2263	-5.2263
3	0.025050	-0.00011	5.0	5.2291	-5.2291
4	0.033400	0.00089	5.0	5.3123	-5.3123

Figure 2. General appearance of the first few rows of the *sample.csv* input file.

The raw plot of the parameter sensing observed in the front-end of the INDe architecture can be seen in **Figure 3**. This is a real-time representation of the internal pressure (strain) and temperature curves as dictated by the freeze-thaw cycle algorithm. Briefly, temperature (blue) is continuously decreased at –2ºC/min until a significant spike in internal pressure surges (black), indicating that a nucleation event has occurred.

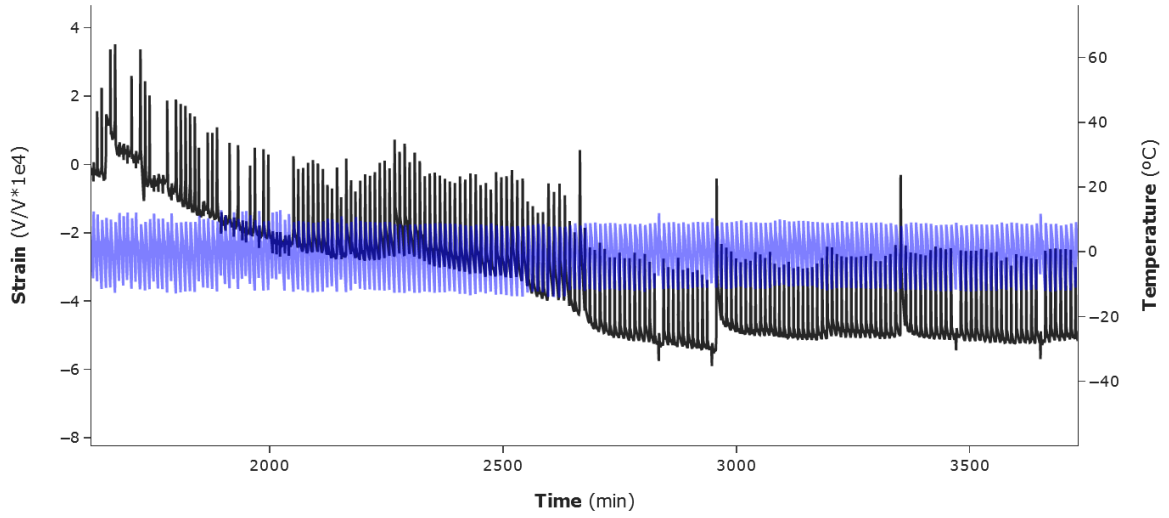


Figure 3. Real-time internal pressure (black) and temperature (blue) sensing of raw data during the freeze-thaw cycle program. *An interactive version and respective code can be found here (Bruno M. Guerreiro 2024 © CC BY-NC 4.0).*

The temperature profile (blue) contains the information of the specific nucleation temperature detected for each cycle, but its determination is an intricate endeavor. In ideal conditions, the nucleation temperature corresponds to the local peak in temperature detected, such that all nucleation temperatures correspond to all local temperature spikes. However, two problems arise in real experimental conditions. First, multiple local extremas are harder to calculate than a single global extreme in a continuous time series, requiring discretization of each cycle. Second, there is an unavoidable sensing lag between the strain gauge sensor detecting a spike in pressure and the freeze-thaw processing unit inverting the cooling phase into a thawing phase, consequently inverting the temperature sensor's telemetry. Therefore, the nucleation temperature data is not collected in the local temperature extremas, but at the temperature T and time t at which a strain spike is detected. Therefore, a mathematical algorithm was designed to locate the accurate nucleation temperatures and plot that information (**Figure 4**).

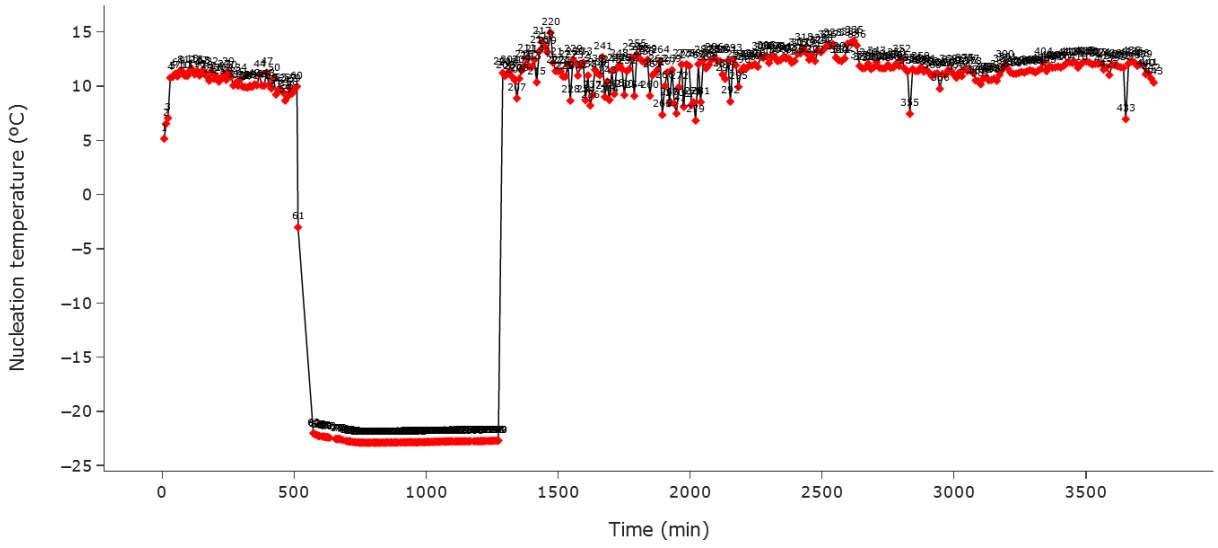


Figure 4. Algorithmic determination of nucleation temperatures for each cycle based on internal pressure local extremas, accounting for physical sensor lag detection. *An interactive version and respective code can be found here.*

In **Figure 4**, each detected nucleation temperature datapoint (red) is labelled by the cycle number to which it corresponds to, as plotted as a continuous time series. These values are collected into a callable Python dataframe (**Figure 5**) which can be updated for the following step.

In [6]:
```
df = pd.DataFrame()
df.set_index = ilocs_max
df['Cycle'] = list(cycle_number)
df['Tnuc'] = list(Nucleation_T)
df['Strain'] = list(Strain_at_NT)
df
```

Out[6]:

	Cycle	Tnuc	Strain
0	1	-5.1498	0.25504
1	2	-6.5347	0.79206
2	3	-7.0608	1.01039
3	4	-10.7624	1.84390
4	5	-10.8480	0.00494
...	...	...	...
438	439	-11.9476	-4.86293
439	440	-11.0938	-5.03369
440	441	-11.1815	-4.69809

Figure 5. General appearance of the first and last rows of the nucleation temperature Python dataframe.

These plots correspond to a total of N=443 cycles collected, yielding an average –0.54±15.83ºC nucleation temperature. One of the reasons automated real-time visualization is important during the processing workflow is to detect non-scientifically relevant data collected, which may arise due to hardware malfunction. In fact, the outstandingly high standard deviation detected by the reporting lines of code and a quick visualization of Figure 4 reveals a period (550–1250 min) in which unknown hardware malfunction occurred, which led to artefactual detection of fictitious nucleation temperatures. Instead of manually deciding on the scientific significance of each nucleation temperature, which would be very time-consuming, a second mathematical algorithm was developed to automatically determine the statistical probability that a given nucleation temperature is scientifically accurate. Briefly, the algorithm calculates the mean and standard deviation based on the provided data and determines a cut-off standard deviation (used as adjustable hyperparameter) which acts as deciding parameter towards significance. If the nucleation temperature of cycle *n* deviates drastically from the nucleation temperature interval democratically perceived by the algorithm, the cycle is not collected into the dataframe, thus adjusting the number of actual cycles. The result is a post-processed cycle list containing scientifically accurate nucleation temperatures (**Figure 6**). After statistical processing, it yielded an average –11.2±1.59ºC nucleation temperature constituted by 304 real cycles.

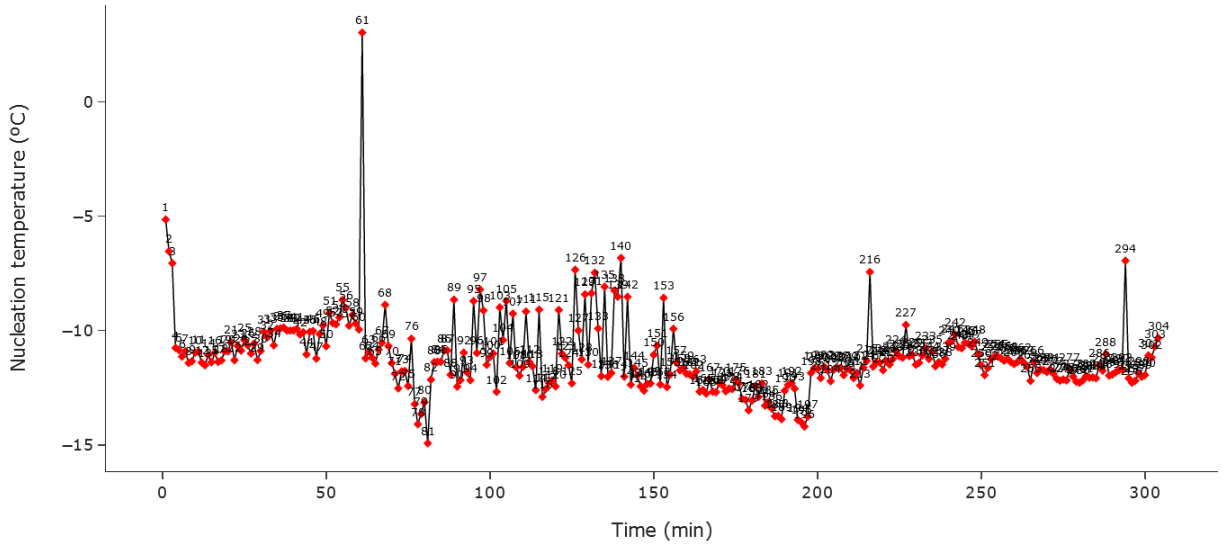


Figure 6. Post-processed nucleation temperature time series, accounting only for scientifically accurate nucleation temperatures. An interactive version can be found here *(Bruno M. Guerreiro 2024 © CC BY-NC 4.0).*

Note that although this algorithm is of static nature and controllable by a user-adjustable standard deviation to define the cut-off, further development can include machine learning to dynamically adapt this hyperparameter based on the molecular system being probed, such that the algorithm *learns* to properly discern between different but scientifically relevant standard deviations. For instance, it was found that FucoPol can narrow the stochasticity of nucleation (the range of nucleation temperatures in the time series) up to a factor of 3 [4]. Therefore, molecular systems like FucoPol differ in their perceived scientifically significant standard deviation relative to pure water. To avoid this, we used lenient rules for validation. In terms of Bayesian statistical testing, this means we opted for flexibility towards type I errors, allowing for some false positives. If a strict standard deviation hyperparameter adapted to FucoPol was implemented to pure water, it would lead to overfitting issues, enforce bias towards type II errors (false negatives) which would lead to valid data deletion and masking of their molecularly distinct behaviors in regulating ice nucleation physics.

A final panel representation of an isochoric nucleation detection experiment running in real-time can be found in **Figure 7**. Therein, three core pieces of information are shown. First, the nucleation temperature time series as a function of cycle elapsed as discussed here, is shown unprocessed.

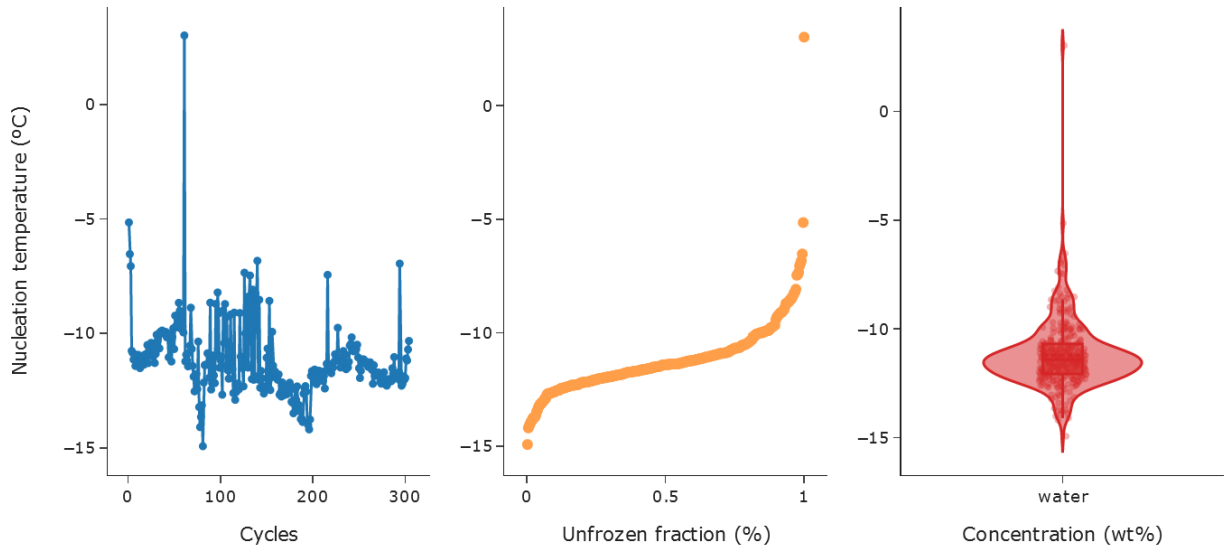


Figure 7. Real-time overview of collected nucleation temperatures for each cycle, unfrozen fraction calculations and nucleation temperature distribution. *An interactive version and respective code can be found here.*

Second, a plot of unfrozen fraction as a function of temperature arises from ordering the nucleation temperatures and modelling them according to a non-homogeneous Poisson process [5], whereby the fraction of unfrozen samples at a given temperature, $\chi(T)$, can be related to the nucleation rate as follows:

$$\chi(T) = e^{\left(-\frac{1}{\beta}\int_{T_m}^{T} J(T)\,dT\right)} = e^{\left(-\frac{\gamma}{\beta}\cdot\frac{(T-Tm)^{1+n}}{1+n}\right)} \quad \text{(Eq. 1)}$$

where T is the temperature, T_m is the equilibrium melting point, β is the cooling rate, and γ and n are empirical fitting parameters. Here, T_m = 0ºC due to a proven non-colligative effect of FucoPol on water thermodynamics [6], and β = 2ºC/min. Lastly, a nucleation temperature violin plot distribution shown in this case for pure water, but can be adapted to compare with other molecules at a given concentration. The data importing and signal processing algorithms can be run simultaneous to the experiment's data collection, enabling the real-time supervision of the system, enabling the troubleshooting of any issues and the quick analysis of the thermodynamic behavior of any type of chemical systems.

2.2. Visualization & automation

The interactive code for Stage 2 of the algorithm is available here (download and open).

After the development of a sample-specific, real-time data processing and analysis algorithm, a global data visualization layout that enables multi-sample comparisons allows to obtain valuable insights in assessing potential scientifically relevant trends. In the work performed, we assessed the changes that the polysaccharide FucoPol, at varying concentration, would generate in isochoric nucleation physics. **Figure 8** presents a side-by-side example of all relevant output plots when comparing the nucleation behavior of pure water and FucoPol. **Figures 8*a–c*** led to the observation that FucoPol could narrow the stochastic range of nucleation temperatures, highlighted in the paper [4] and further highlighted in a different perspective from the violin distributions in **Figure 8*e***. In **Figure 8*f*** it is further emphasized that nucleation temperatures can be further segmented and analyzed if bimodal trends ever arise in the distributions, the algorithm thus also being sensitive to the shape of data distribution. Likewise, unfrozen fraction plots (**Figure 8*d***) are quintessential for the Poisson modelling in the next stage of data analysis. The visualization stage not only provides a real-time overview of the experiment, but also yields publishable graphics instantaneously from the data being collected in real time. The continuous gathering of isochoric nucleation data and the nature of data storage algorithms eventually provides the basis for database generation, from which quick visualizations and comparisons can be performed over a wide variety of molecules and experimental conditions. This database can be shared publicly and potentiate predictive analysis amongst the scientific community.

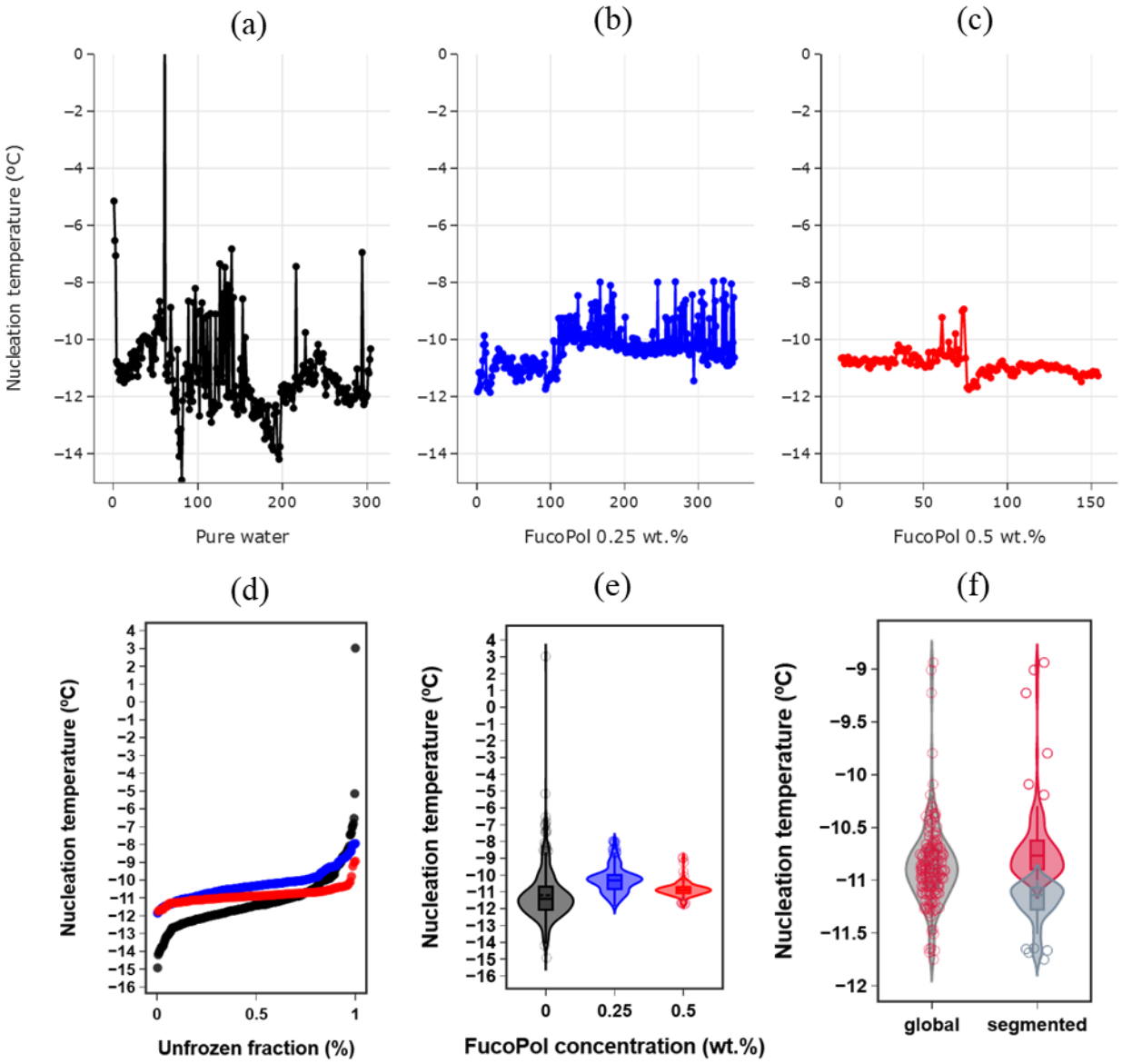


Figure 8. Data visualization overview of the isochoric nucleation experiment being run in-real time for the 0.5 wt.% FucoPol solution. The interactive plots are updated at every datapoint collected in (c), while a quick comparison with previous experiments using (a) pure water and (b) 0.25 wt.% FucoPol are shown side-by-side. In the bottom panel are represented simple data transformations that calculate the unfrozen fraction slopes using Eq. 1 (d), and the nucleation temperature plot distributions (e) and (f) derived from the data processing discussed previously. *An interactive version and respective code can be found here (Bruno M. Guerreiro 2024 © CC BY-NC 4.0).*

2.3. Poisson iterative modelling

The interactive code for Stage 3 of the algorithm is available here (download and open).

The last stage of isochoric nucleation data handling revolves around data analysis and modelling. The novel approach used in this paper for interpreting nucleation temperatures was to rationalize the nucleation event as a Poisson event materializing in time. Therefore, the Poisson modelling shifts interpretation of nucleation from the temperature-domain to the time-domain and can yield valuable information into how much time may elapse until the first critical nucleus forms (induction time), thus yielding a metric of cryopreservation system stability from a practical standpoint.

First, to interpret how long an isochoric system will remain stable and ice-free before the first ice nuclei form, the nucleation rate J(T), which is the number of critical size nuclei formed per unit time, may be estimated by the following power-law:

$$J(T) = J_0 \times e^{\left(-\frac{\Delta G_n}{kT}\right)} = \gamma \Delta T^n \qquad \text{(Eq. 2)}$$

where J_0 is a constant, ΔG_n is the activation energy barrier for nucleation, k is the Boltzmann constant and T is the temperature, γ and n being empirical parameters. The Python code for Eq. 2 is presented in **Figure 9**.

```
In [78]: import math

def poisson_model(T, gamma, n):

    ### CONSTANTS ###
    A = 2.9
    beta = -2
    Tm = 0

    # division of constants
    const = A/beta
    # numerator
    num = np.multiply(gamma, np.power(np.subtract(T, Tm), 1+n))
    # denominator
    denom = 1+n
    # do division
    tmp = const * (num/denom)
    # exponentiate
    return tmp
```

Figure 9. Python script for defining a function that computes the nucleation rate J(T) through a non-homogenous Poisson model. Then, several iterations of the function will establish the theoretical induction time curves.

Then, from a subset of nucleation temperatures for each sample, the experimental unfrozen fraction $\chi(T)$ is calculated from the `poisson_model()` function. The Poisson model fitting is split into two stages: (i) a first-approximation fit, which uses default static values for each variable (**Figure 10*a***); (ii) an iterative optimization of the empirical parameters until optimal R^2 and RMSD metrics are achieved (**Figure 10*b***).

During the first-approximation fit, the relationship between both variables is quite poor as expected. In order to minimize the amount of iterations for optimal fit, the algorithm selects the centermost datapoints in the distribution, while maximizing the amount of datapoints chosen. This selection is physically relevant, and derives from the common rationale employed in nucleation physics and atmospheric sciences that any nucleation temperatures outside the 10-90% distribution range can be considered rarely occurring nucleation events of very low probability [7], thus not accurately representing the nucleation behavior of the system. Attempting to fit extreme values to the Poisson fitting would lead to a trade-off of overfitting all values instead of optimally fitting relevant temperature ranges.

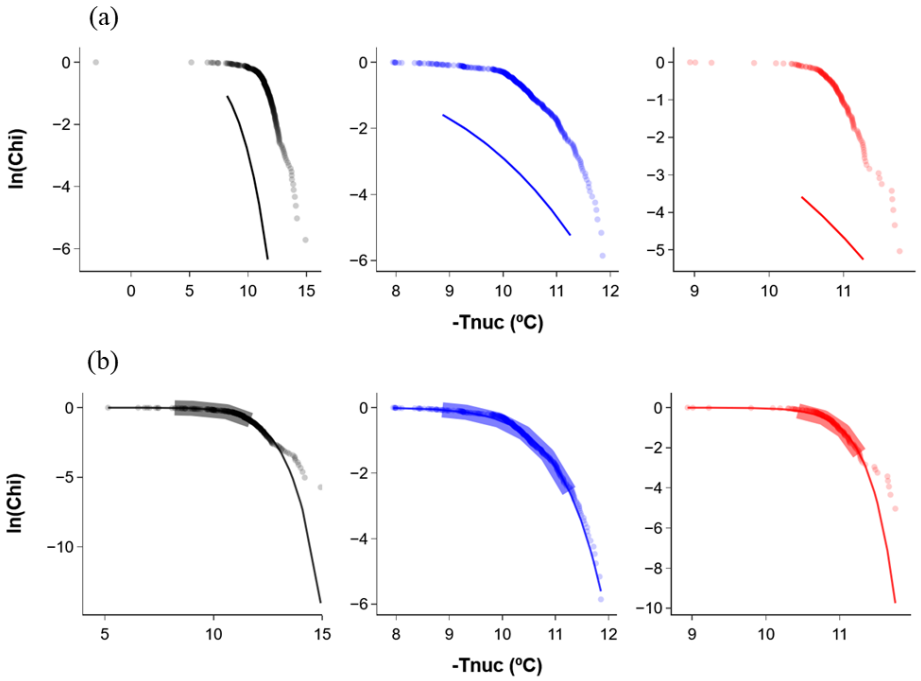


Figure 10. Application of the Poisson model to the experimental isochoric nucleation data, by plotting the unfrozen fraction χ as a function of experimentally recorded nucleation temperature. (a) First-approximation fit using empirical parameters. (b) Optimized fit after n=12000 iterations. The shadowed region represents the best-fit interval of values to describe the unfrozen fraction slope of the system. *An interactive version and respective code can be found here.*

The iterative optimization of the empirical parameters γ and n follows a sequence of coding blocks which are described below:

1. The root mean squared error (RMSE) metric between theoretical and experimental $\chi(T)$ values is generated for each sample.
2. Using the `scipy` Python library, the `curve_fit()` function is used to calculate `popt` and `pcov` metrics for each experimental and theoretical data pair. `popt` is an array which contains the optimal values for the parameters so that the sum of squared residuals is minimized. `pcov` is an estimated approximate covariance of `popt` and is used later for the determination of standard deviations.
3. Standard deviations σ^2 were computed from the optimal `popt` and `pcov` values.
4. The average correlation coefficient R^2 was computed for all samples.

After 12000 iterations, the empirical parameters γ and n were established for each sample, with attributed standard deviations, and an optimal R^2 correlation factor for the fitting. The theoretical fit is shown superimposed to the experimental data in **Figure 11**, highlighting the validity of Poisson distributions correlating well with ice nucleation experiments. The empirical parameters obtained through Poisson modelling, given optimal fit, serve as unique identifiers for each molecular system and can establish intrinsic characterization parameters for their unique isochoric nucleation behavior.

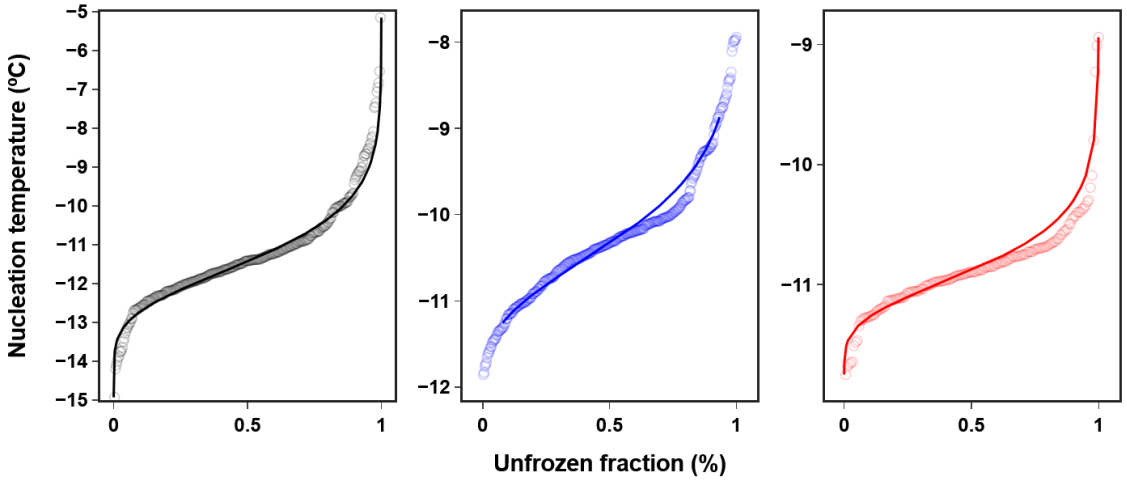


Figure 11. Poisson theoretical fitting superimposed to the experimental isochoric nucleation temperature data for pure water (black, γ=1.0×10^{-13}, n=8.0, R^2=.942), 0.25 wt.% FucoPol (blue, γ=2.5×10^{-29}, n=26.8, R^2=.881) and 0.5 wt.% FucoPol (red, γ=8.2×10^{-38}, n=34.3, R^2=.918). *An interactive version and respective code can be found here.*

2.3.1. Stability map plotting on PyCharm

As stated before, the Poisson-based conversion of nucleation data from the temperature-domain to the time-domain yields practical insight into how long a system will remain in stable supercooled state before the first super-critical ice nucleus emerges. By definition, this was coined as induction time, τ, is plotted in **Figure 12*a***, and is inversely proportional to the nucleation rate, $J(T)$, as follows:

$$\tau = J(T)^{-1} \quad \text{(Eq. 3)}$$

Figure 12*a* can be found on stage 3 of the algorithm presented in the last section, but further transformations using PyCharm can yield highly informative time stability maps (**Figure 12*b***). To generate this contour plot, a Python script for creating a graphical user interface (GUI) using the `tkinter` library was adapted for this specific purpose (**Figure 12*c***). Briefly, the algorithm fits mathematical (including the Poisson) models to the temperature-concentration relationship. In order to express timescales in the contour plot (year, month, week) as delimiters of supercooled state stability, the algorithm facilitates the mathematical transformation of a 3-dimensional matrix of data (three variables) into multiple 2D contour plots as a function of induction time in the *z*-axis.

Time stability maps inform the cryobiologist how the choosing of critical parametric combinations may yield an ice-free system for a given timescale. These decisions are statistically driven due to the probabilistic nature of the Poisson modelling and nucleation itself. However, the complete thermodynamic understanding of nucleating systems and its interpretability through **Figure *12b*** allows to design biopreservation scenarios under safe temperature, solute concentration and duration of preservation ranges to maximize biopreservation success. An example of this intuitive rationale is the following: pure water in an isochoric system is expected to remain in a metastable supercooled state for about 6 months at –5ºC until nucleation initiates, but only 5 hours at –8ºC. The addition of 0.25 wt.% FucoPol at –8ºC increases system stability to two weeks, or more than a year with 0.5 wt.% FucoPol, which exponentially increases the security of preserved biological matter.

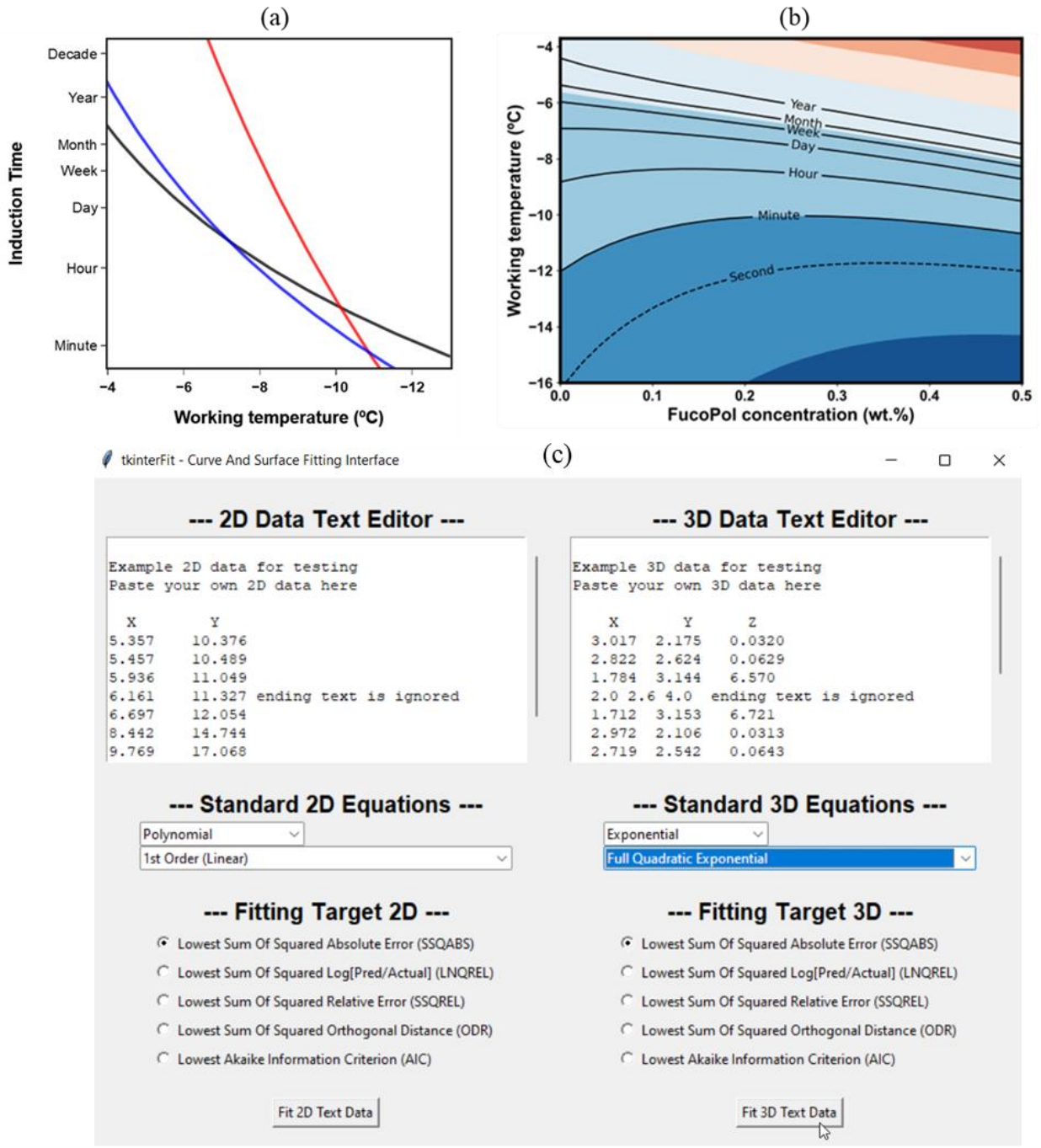


Figure 12. Induction time stability map generation using a combination of Poisson modelling, isochoric nucleation data and contour plot matrix transformations in PyCharm using the `tkinter` GUI. *The respective Python script can be found here.*

2.4. How did using Data Science solve this problem?

Without knowledge in Python programming, processing algorithms and data science analytical tools, the interpretation of isochoric nucleation data would be very time consuming and of limited insight. In this complete workflow, all these stages of development are interconnected and fully automated. From the first nucleation temperature recorded to the induction time stability map, the Python workflow is updated and visualized in real-time, allowing for high-throughput data analysis and decision-making. Although a freeze-thaw cycle can be obtained in a minute, the statistical validation of a system requires at least 300 cycles, which takes about 12 hours, and the manual analysis and insight generation from all collected data could take weeks. Using this automated Python workflow, which confers a proper standardized way of collecting data, not only yields a publishable thermodynamic report that can be obtained in seconds once the final datapoint is collected, but can be used to build isochoric nucleation comparative databases using a standardized methodology.

Application #2: Principal Component Analysis revealed a distinct fucose mechanism in cryopreservation

For several decades, one of the most fundamental tenets in cryobiology has been that molecular charge is the main predictor of ice growth disruption, and therefore, cryoprotective activity [8]. The ability of a molecule to disrupt ice growth is essentially linked to its ability to mimic hydrogen bond donor-acceptor dynamics [9]. Based on these fundamentals of molecular interaction, any molecule with a permanent charge dipole can interact with freezing water and disrupt its molecular assembly and the directionality of hydrogen bonds necessary for robust ice to form. The validity of this tenet has been corroborated consistently over the years for small molecule polyols [10], [11], antifreeze proteins [12]–[14] and several ice recrystallization inhibitors [15], [16]. It then follows for polysaccharides that maximizing net formal charge should maximize cryoprotective function. However, in a recent broad screening of 26 polysaccharides of variable composition, molecular weight and net formal charge, in our paper titled "Fucose is an essential feature in cryoprotective polysaccharides" [17], this has not been observed. Rather, a statistically significant predominance of electronegative polysaccharides that are fucose-rich consistently dominate the highest tiers of cryoprotective performance. Fucose demonstrated a critical role in enhancing cellular cryopreservation outcomes, but because it is a neutral monomer, it suggested that an extension of the scope of the polyanionicity tenet is now justified. Moreover, this enhancement in function was not observed for other neutral monomers in the composition. Thus, the mechanism by which fucose acted remained unknown.

3.1. Principal Component Analysis

Due to the multiparametric and convoluted nature of this study, we performed a data analysis method called Principal Component Analysis (PCA). PCA allows to reduce the number of dimensions (predictor variables) contributing to the variability in the final outcome (cell survival) by grouping variables behaving similarly into principal components (PCs), which possess a pattern

of contribution towards variability. Each PC, although a combination of variables which might not inter-relate biologically, hints at probable biological mechanisms that may be distinguishable from common factors amongst the variables constituting a given PC. To apply this methodology, a Python script was developed to perform PCA to a dataset of 26 bio-based polysaccharides (**Table 1**), composed of defined quantities of uronic acids (wt.%), fucose (wt.%), sulfates (wt.%), acyl groups (wt.%) and average molecular weight (MDa). *The Python script can be found here.*

Table 1. Overview of the *input.csv* file used in the PCA algorithm containing experimental data. Each row corresponds to an indexed polysaccharide in the array. PTV stands for post-thaw viability, an indicator of cell survival after the cryopreservation procedure. UA: uronic acids, MW: molecular weight, Pyr: pyruvate, Suc: succinate.

Outcome variable	*Predictor variables*						
PTV	**Fucose**	**UA**	**Neutrals**	**MW**	**Acyls**	**Pyr+Suc**	**Sulfate**
2.91	34	9.5	58	3.75	20	16	0
2.15	0	0	31.1	0	1.4	0.5	6
1.299	0	42.11	42.11	3.1	5.4	4.9	2.8
1.329	0	45.45	54.55	4.6	2.5	2	3.3
2.172	4.35	17.39	47.83	1.2	1.1	1.1	3.4
0.852	0	45	40	1.4	2.1	0	2
2.127	0	22.22	44.44	3.4	1	0.5	2.9
2.719	37	23	40	4.2	9.9	2.85	0
2.935	0	37.5	50	3.2	6.2	5.5	3.4
2.444	0	50	0	0.513	0	0	0
3.069	26.38	9.29	64.33	7.88	10	0	0
0.954	1	11.7	34.6	0	0	0	10.6
3.01	45.04	7.09	47.88	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Briefly, the experimental data of **Table 1** was used as input.*csv* file, stored in a variable called `data` and its contents standardized. First, scale normalization of each variable was carried out. Variable scaling is important because the PCA algorithm does not distinguish orders of magnitude, such that a molecular weight datapoint of 1 000 000 Da would have a heavier contribution towards variability than 1 MDa, although they are identical. Then, the `PCA()` function from the `scikit-learn` library was performed, defining the calculation to address a total of six principal components (the total number of PC should not be higher than the total number of variables). The Python script used for PCA implementation in this dataset is as follows:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Input raw data
data = pd.read_csv('PCA.csv', index_col=0)

# Standardize the data
scaler = StandardScaler()
data_std = scaler.fit_transform(data)

# Perform PCA
pca = PCA(n_components=6)
principal_components: object = pca.fit_transform(data_std)
explained_variance_ratio = pca.explained_variance_ratio_
pcnumber = ['PC1', 'PC2', 'PC3', 'PC4','PC5', 'PC6']

# Create a dataframe to store the results
results = pd.DataFrame(data=principal_components, columns=pcnumber)
results['Sample'] = data.index
print(explained_variance_ratio) #prints the variance for each PCx

# Visualize the results
plt.figure(figsize=(8, 6))
plt.scatter(results['PC1'], results['PC2'])
plt.xlabel('PC1 ({:.2f}%)'.format(explained_variance_ratio[0]*100))
plt.ylabel('PC2 ({:.2f}%)'.format(explained_variance_ratio[1]*100))
plt.title('PCA of Cryoprotective Polysaccharides')
for i, sample in enumerate(results['Sample']):
    plt.annotate(sample, (results['PC1'][i], results['PC2'][i]))
plt.show()

# Create a scree plot
plt.figure(figsize=(8, 6))
plt.plot(np.arange(1, len(explained_variance_ratio)+1), explained_variance_ratio, 'o-')
plt.xlabel('Number of Principal Components')
plt.ylabel('Explained Variance Ratio')
plt.title('Scree Plot of Cryoprotective Polysaccharides')
plt.show()

# Loading plot
loadings = pd.DataFrame(pca.components_.T, columns=pcnumber, index=data.columns)
plt.figure(figsize=(8, 6))
plt.scatter(loadings['PC1'], loadings['PC2'])
plt.xlabel('PC1 Loading')
plt.ylabel('PC2 Loading')
plt.title('Loading Plot of Cryoprotective Polysaccharides')
for i, feature in enumerate(loadings.index):
    plt.annotate(feature, (loadings['PC1'][i], loadings['PC2'][i]))
plt.show()

# Cluster the samples
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(principal_components)
results['Cluster'] = kmeans.predict(principal_components)
```

```
# Get centroids
centroids = pd.DataFrame(kmeans.cluster_centers_, columns=pcnumber)
centroids['Cluster'] = ['Cluster {}'.format(i+1) for i in range(len(centroids))]

# Plot the results
fig, ax = plt.subplots(figsize=(8, 6))
colors = ['#40A599', '#C9505E', '#000000'] # list of colors for each cluster
for i, cluster in enumerate(set(results['Cluster'])):
    mask = results['Cluster'] == cluster
    ax.scatter(results.loc[mask, 'PC1'], results.loc[mask, 'PC2'], c=colors[i], label=cluster,
alpha=.75)
    centroid = centroids.loc[centroids['Cluster'] == cluster]
    ax.scatter(centroid['PC1'], centroid['PC2'], c='black', s=100, marker='x', facecolors='black')
for i, sample in enumerate(results['Sample']):
    plt.annotate(sample, (results['PC1'][i], results['PC2'][i]), size=8)
ax.legend()

# Add dashed lines where x=0 and y=0
ax.axhline(y=0, linestyle='dotted', lw=1, color='grey')
ax.axvline(x=0, linestyle='dotted', lw=1, color='grey')

# Add loading vectors
loadings = pd.DataFrame(pca.components_.T, columns=pcnumber, index=data.columns)
for i, feature in enumerate(loadings.index):
    plt.arrow(0, 0, loadings['PC1'][i]*2, loadings['PC2'][i]*2, color='r', alpha=.5, linewidth=2,
head_width=.02, head_length=.02)
    plt.text(loadings['PC1'][i]*2, loadings['PC2'][i]*2, feature, color='r', alpha=.7)

# Set axis labels and title
plt.xlabel('PC1 ({:.2f}%)'.format(explained_variance_ratio[0]*100))
plt.ylabel('PC2 ({:.2f}%)'.format(explained_variance_ratio[1]*100))
#plt.title('PCA Biplot with Clusters and Centroids')
ax.legend().remove()

# Show the plot
plt.style.use('fivethirtyeight')
plt.savefig('biplot.tif', dpi=600, format='tif')
plt.show()
```

The final output plots of the PCA method are summarized in **Figure 13** and will be addressed point-by-point alongside the corresponding code blocks and the interpretations of its results. First, the two principal components with highest contribution towards post-thaw viability (PC1, 36.89% and PC2, 30.25%) are shown in a 2D scatter plot (part of **Figure 13*e***), where each labelled datapoint corresponds to a tested polysaccharide and its corresponding cellular post-thaw viability. In the preliminary plot, only datapoints are shown and some clustering can be identified. However, the only information conveyed is the positioning of each datapoint relative to the *x* and *y* intercepts which constitute the origin of the scatterplot (PC1 = PC2 = 0). In essence, the more biased towards a specific principal component a datapoint is, the more exclusive it is to that component. For instance, blue datapoints constitute almost exclusively PC1, with small deviations towards the horizontal intercept of PC2. From this information alone, PCA is a very powerful method for

determining contributions towards variability, because a simple dimensionality reduction from 6 variables to 2 components has led to visible data clustering that can be traced back to their original polysaccharide labels and their chemical composition.

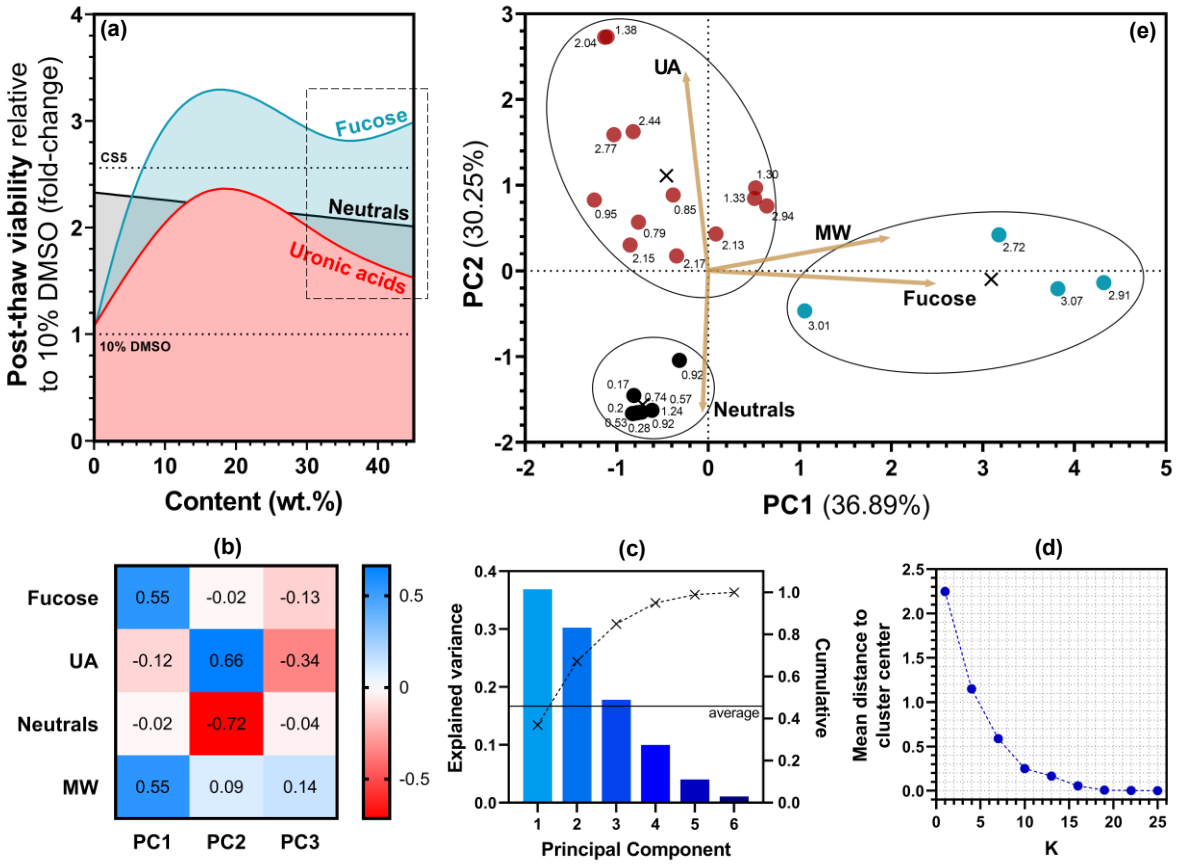


Figure 13. Summary of multivariable effect deconvolution in cryoprotective polysaccharides. (a) QSAR optimal compositional spectrum showing the optimal range for fucose and UA content, and comparative groups: control (10% DMSO) and commercial cryogenic formulation (CryoStor™ CS5); (b) Correlation matrix, quantitatively describing the contribution of each variable to a principal component. (c) Scree plot of explained variance (bars), complemented by the cumulative explained variance for all principal components (dashed line). (d) K-means clustering score model that iteratively calculates the mean distance of each datapoint to N cluster centers and returns an optimal K clusters; (e) PCA biplot with eigenvector loadings for each predictor variable and point clustering. The arrows that extend outward from the origin are eigenvectors, which indicate the variance loading that a variable has on each principal component. The degree of influence that each element has on a principal component is visualized by the angle and length of the arrow relative to the axes. *The Python script can be found here.*

Then, the Screen plot shown in **Figure 13*c*** shows how much variance in the outcome variable can be explained by each Principal Component. The dashed line represents the cumulative explained variance, such that 100% of the variability can be explained by all components, and the horizontal line corresponds to the average explained variance. Given that only PC1 and PC2 constitute an explained variance that is statistically greater than the geometric average, only these two were chosen for further evaluation and real variable traceback. Any attempts at choosing more than two Principal Components (in this experimental scenario) would lead to statistical bias and overfitting of the model, yielding less robust insights.

The correlation matrix shown in **Figure 13*b*** further demonstrates which variables are constitutive of each principal component, and such a correlation is vectorized in **Figure 13*e***. For instance, fucose shows an eigenvalue loading of 0.55 towards PC1 and near zero towards PC2. This means that the contribution of fucose towards post-thaw variability can be solely explained through the contribution of PC1. Lastly, **Figure 13*d*** reveals the *K*-means clustering algorithm implemented to visualize clusters of data in **Figure 13*e***. Essentially, the clustering algorithm computes the mean distance of each datapoint to an empirical centroid value in the cartesian plane, and by iterative minimization of such distances, the algorithm reveals how many clusters of data may exist. In this case, three clusters were identified: the first cluster is composed of polysaccharides demonstrating a high cryoprotective performance due to their contents in uronic acids. This is corroborative of the polyanionicity tenet previously discussed. A second cluster contains low performing polysaccharides for which the neutral monomer content is dominant. As expected, lack of polyanionicity leads to lack of cryoprotective function, and is axially opposed to the uronic acid presence, as part of the same principal component (PC2). This is highly suggestive that PC2 group polysaccharide properties exert a contribution towards cell survival by a charge-dependent mechanism. Likewise, a third cluster, uniquely loaded by fucose and molecular weight contributions towards PC1, was identified. A deeper analysis reveals that these are high-performing polysaccharides which also contain uronic acids in their composition (first cluster), but distinguish themselves through an additional high content in fucose, which drastically enhances performance. Likewise, PC1 then correlates to a charge-independent mechanism, because it is mostly composed of fucose influence and is orthogonal to PC2. Based on the cell recognition properties of fucose [18] and the known importance of cell volume regulation during cryopreservation [19], polyanionic fucose-rich polysaccharides were hypothesized to also interact

and indulge in cell membrane stabilization. This enhancement of function with the concomitant presence of hydrophilic uronic acids and hydrophobic fucose spacers is consistent with recent theories of an hydrophilic-phobic balance being paradoxically essential towards function [20].

3.2. How did using Data Science solve this problem?

Without the implementation of **Principal Component Analysis** through **PyCharm** using the **Python programming** language, the convoluted nature of the influence of chemical composition in cryoprotective outcome would remain unresolved. Although a better understanding of the system benefits from limitlessly increasing the number of polysaccharides studied, such a scenario is limited by experimental screening. An increased number of dimensions may also lead to increased and more complex insights but risks misinterpretations due to potential correlations between variables that are not isolated enough into individual variables (leading to collinearity issues). Overall, the discovery that fucose plays a critical role in cryoprotection did not remain limited to its presence being visibly beneficial but was attributed to a mechanism other than that regulated by uronic acids, which indicates that it has a chemically distinct behavior. Moreover, the fact that the fucose eigenvector aligns with the molecular weight eigenvector suggests that their contribution towards an increase in cryoprotective outcome are similar. An increased molecular weight is often associated with the intrinsic property of enhanced viscosity and molecular entanglement. Likewise, an increase in fucose content may lead to increased cumulative cell membrane interactions, resolving into the intrinsic property of beneficial cell membrane integrity regulation.

Application #3: Dual nucleation behavior elucidated by Classical Nucleation Theory interactive energy landscapes

4.1. The problem: incongruent simultaneous dual nucleation behavior

In Application #1, the nucleation temperature distribution plots for FucoPol solutions (**Figure 8**) revealed an interesting trend. With an increase in concentration, the narrowing of the spread of nucleation temperatures became increasingly asymmetric. In other words, an extinction of supercooled nucleation temperatures was more pronounced than nucleation temperatures above the average value (Figure 14). This statistical skewing of the nucleation temperature spread led to interpretations regarding its physical origin, but regardless of narrowing magnitude, a dual anti-nucleation (red) and pro-nucleation (green) effect was acknowledged.

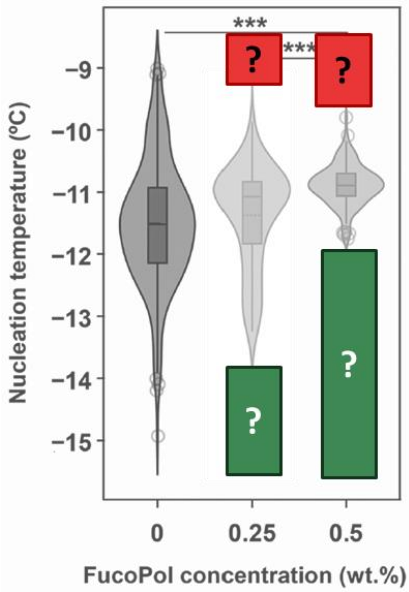


Figure 14. Nucleation temperature distributions for pure water and FucoPol aqueous solutions. An increasingly asymmetric stochastic narrowing is observed with increasing FucoPol concentration, prompting discussions regarding its thermodynamic nature.

Conceptually, the anti-nucleation effect was interpreted as an extinction of nucleation at higher temperatures during cooling, leading to a delay in phase transition probability. Conversely, a pro-nucleation effect, rather than pointing to anticipated nucleation in the time domain, hinted at increased probability of nucleation occurring at temperatures closer to the average value than supercooled temperatures (near –11ºC rather than in the –12 to –15ºC range). This increased probability is a trivial inference from observation, because for equal volume and cooling rates, the system will undeniably undergo a phase transition. However, an anticipation of nucleation in the temperature-domain must be reflected thermodynamically in a change of the energetic landscape of the system. The problem with justifying a dual nucleation behavior was that, by using Classical Nucleation Theory (CNT) models, a single energetic formalism could not validly accommodate both effects.

4.2. The solution: plotting CNT energetic landscapes in Streamlit.io

To reconcile the simultaneous effects of nucleation anticipation and delay, the widely accepted CNT energetic profiles were plotted interactively in the front-end Python web application Streamlit, which allows for fast visualization of the Python scripts produced specifically for this application. *The interactive web application can be found here.* **Figure 15** shows the energetic landscape of a pure water system, only prone to homogenous nucleation, reflecting the change in Gibbs free energy ΔG as a function of cluster (nuclei) radius r. The energy barrier for nucleation to occur and evolve past unstable clusters into stable ice crystals can be modelled as a gain-cost function (green), where a trade-off exists between favorable volumetric free energy (blue) and a counteracting interfacial energy (red). The volumetric free energy ΔG_v is proportional to the number of atoms present in a cluster. Therefore, a bigger cluster size will facilitate the migration of free water molecules into the cluster, in a continuous positive feedback loop until the cluster achieves a radius r that allows it to grow into a stable nucleus, rather than continuously re-dissolving back into solution. The interfacial energy γ is a reflection of nuclei geometry and describes the potential energy required to enlarge an interface. If the shape of a nuclei allows for free molecular docking without dominant repulsion, interfacial energy is low, and the energy barrier for nucleation to occur is reduced, enhancing nucleation probability.

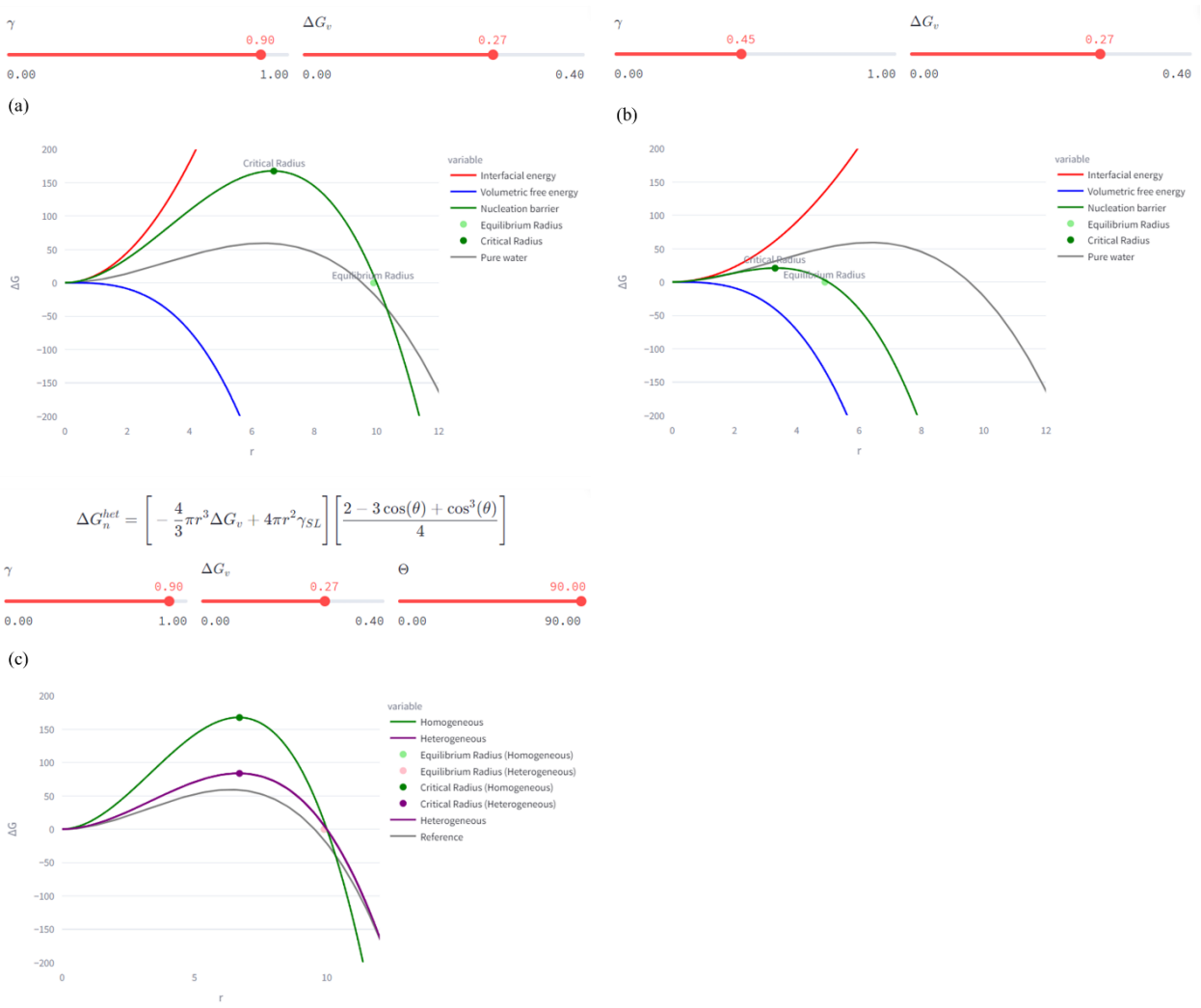


Figure 15. Energetic landscape of nucleation phenomenology, according to Classical Nucleation Theory. In gray, the theoretical homogenous nucleation of pure water is shown for reference. In (c), the constraints for heterogeneous nucleation are subsequently embedded in the formalisms. *The interactive visualization can be found here.*

The interactive visualizations using Streamlit.io allow to finetune contributive physical parameters, such as interfacial energy γ and volumetric free energy ΔG_v to hypothesize potential differences in the energetic landscape. The main advantage of using interactive visualization is the ability to quickly assess the validity of hypothetical system conditions without needing to formally characterize the thermodynamic properties of the system, or the accurate mathematically constraints to CNT formulations, in order to produce a given energetic landscape. **Figure 15** therefore represents an extreme-value exploration of the γ and ΔG_v and their influence in the

experimental nucleation barrier (green). For instance, for a constant-volume system characterized by constant ΔG_v, the high interfacial energy in **Figure 15*a*** demonstrates an increased difficulty in nucleation occurrence, due to an increased nucleation barrier. However, a 2-fold reduction in interfacial energy (**Figure 15*b***) leads to a drastic reduction in the nucleation energy barrier, below that of pure water. This would be indicative of a pro-nucleation effect, relative to pure water conditions. Moreover, the critical radius r^* required for nucleation to occur is also reduced, concomitant with the equilibrium nuclei radius, which is the average dimension of cluster radii, when rationalized as a Gaussian distribution of nuclei sizes. In conclusion, a reduction in interfacial energy facilitates nucleation at lower nuclei sizes.

By increasing the number of contributive parameters and plotting additional metrics, such as the Zeldovich factor, thermal fluctuations near the critical radius r^* and spread of nuclei radii near ΔG^*, one can find the conditions for which a probable energetic landscape may explain the dual pro/anti nucleation effect observed for FucoPol (**Figure 16**).

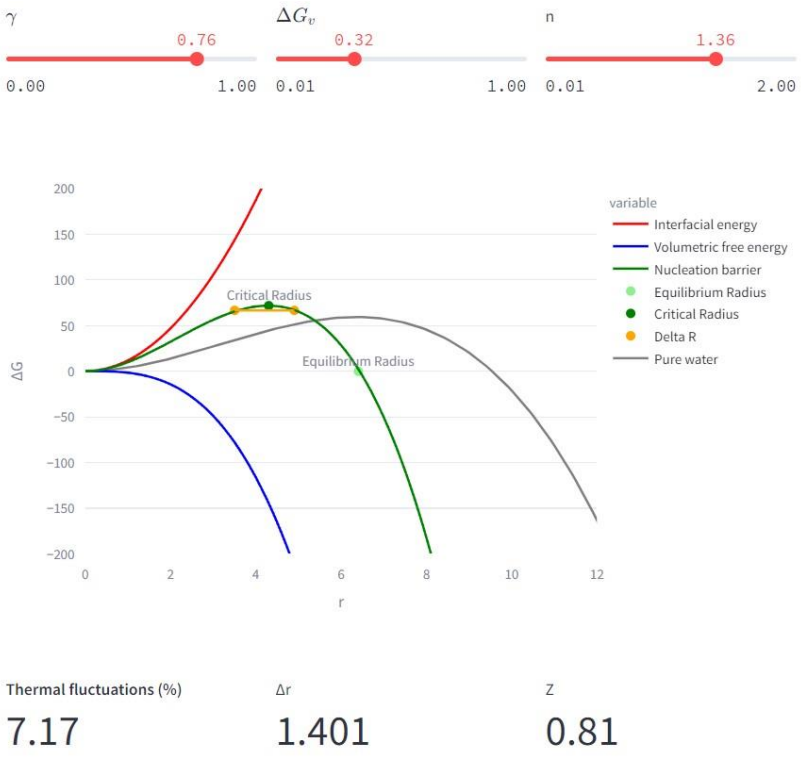


Figure 16. Final energetic landscape visualizations in Streamlit.io, with relevant comparative CNT metrics. *The interactive visualization can be found here.*

The energetic scenario depicted conforms to an anti-nucleation effect as observed in Figure 14, due to slightly increased nucleation barrier, but also implies that the critical radius required for nucleation to occur (once it occurs) is lower, and results in the formation of smaller-sized nuclei, by concomitant reduction of the equilibrium radius. The latter observations would be consistent with a pro-nucleation effect. However, this can only occur if the total amount of water molecules n in the system changes, which is incompatible with a constant-volume system.

Figure 16 allowed to conclude that an inherent change in system conditions must exist to accommodate the existence of a PRO/ANTI nucleation behavior, as simultaneous effects are incongruent with a single CNT formalism. At the onset of cooling, a gradual increase in system viscosity hinders the diffusional mobility required for hastened nucleation to occur, leading to an apparent delay in nucleation, and therefore, an anti-nucleation effect. However, the viscosity effect alone does not explain the extinction of nucleation at lower temperatures. From previous research on FucoPol rheology, calorimetry and circular dichroism, it is known that the polysaccharide undergoes a sol-gel transition, forming a gel matrix of defined porosity (*under review*). A stochastic narrowing of nucleation temperatures implicitly indicates that the system is more deterministic. In thermodynamic conceptualization, an increased determinism in consistently obtaining similar nucleation temperatures indicates that the initial conditions during pre-nucleation must be more uniform. In the presence of a gel-state, which naturally assembles a matrix of sensibly uniform and equal sized pores (which is in itself an identifying property of each molecular system), it becomes probable that an equal amount of water molecules is available in each pore. Nucleation is intrinsically sensitive to system scale, such that milliliter amounts of pure water will nucleate differently than liter amounts.

Although the anti-nucleation effect could be explained by viscosity enhancement and the nucleation temperature narrowing is plausible when pore formation is considered, the pro-nucleation effect is only rationalizable if a sol-gel transition truly exists. The extinction of nucleation at lower temperatures than the average T_n was hypothesized to be a survival bias generated by volumetric confinement. In other words, the fractioning of the bulk water volume available to freeze into smaller fractions split between each pore reduces the system scale n by an unknown factor. Naturally, and by interactive introspection of the CNT landscape, a reduction in n leads to a decrease in interfacial energy by a factor of $n^{2/3}$ which outweighs the change in

volumetric free energy (scales with $-n$). In other words, the nucleation barrier is flatter, which increases the probability of nucleation to occur (pro-nucleation effect), to both small and large cluster sizes. However, in small pores containing a limited amount of water molecules, the radius r of formed nuclei can only grow to an extent, effectively reducing the critical nuclei radius r^* that is achievable. This survivorship bias is in in accordance with the smaller crystals observed in the presence of FucoPol [6]. Lastly, this survivorship bias, which generates an apparent disposition towards favoring smaller crystals to form, did not explain if a kinetic acceleration of crystal growth itself was taking place. From calorimetric data, an anticipation of the freezing point is factually present, but there is no evidence of growth acceleration from directional freezing experiments [21]. Therefore, the anticipation of nuclei and crystal growth regarded as pro-nucleation and pro-crystallization effects are not related to kinetic enhancement of the thermodynamic phenomenon, but an increase in phenomenological probability.

4.3. How did using Data Science solve this problem?

Although the dual nucleation behavior observed allowed for establishing a hypothesis for the potential mechanisms in action during cooling, the interactive analysis of energetic landscapes using **Python programming** and **Streamlit web applications** proved fundamental to enable the justification of those propositions using widely accepted thermodynamic models, without access to accurate mathematical restraints or complete knowledge of the molecular system during isochoric supercooling. Without a visual understanding of how the nucleation barrier landscape can determine the observed nucleation distributions, it would be implausible that a change in system conditions as hypothesis could be stated in simpler, more communicable terms. Lastly, the understanding that a reversible change in system chemistry is necessary to justify the observed duality later became congruent with the sol-gel transitions observed for cryoprotective polysaccharides, an exploration incited by these discoveries.

Application #4: Multidimensional meta-analysis and extremophilic polysaccharide database generation

The ability to predict if a polysaccharide may possess cryoprotective function remains a highly complex and challenging problem. Based on a thorough literature search, a database of 145 polysaccharides produced from extremophilic microorganisms was compiled, in an attempt to discern major differences in molecular composition, conformation and functionality based on the natural adaptation to different habitats. This database contains 128 organic and 16 mathematically calculated parameters, for a total of 144 parameters (or dimensions), and a meta-analysis of its contents is currently being drafted, pending invitation for a special issue in *New Biotechnology*. The database was split into categories, with 12 variables characterizing microorganism identity, 22 variables for microorganism growth/EPS production conditions, 33 for polysaccharide composition, 12 for polysaccharide structure, 10 for EPS macromolecular fractions, 33 for polysaccharide characteristics (15 for physicochemical properties + 18 for biological functions) and 14 for cryoprotection (7 for biological evidence + 7 for explanatory mechanisms of action) as outcome function. **Figure 17** shows a surface-level overview of the database, with helpful visualizations which allow to explore deep insights in the data. *The interactive visualizations can be found here.*

From exploratory data analysis, several trends were collected from multidimensional analysis. To briefly name a few: *(i)* hydrogen bonding and electrostatic forces dominate bioactivity in polysaccharides, due to the prevalence of uronic acids; *(ii)* marine bacteria contain oddly but consistently high amounts (20-50%) of uronic acids in its polysaccharide content; *(iii)* conformational parameters revealed helical polysaccharides adapt better to temperature variations due to reversibility, flexibility, and preservation of rheology; *(iv) t*he crucial polyanionicity of psychrophiles (cold-adapted) is also present in halophiles, indicating that cryoprotection and osmoprotection are common denominators of damaging volumetric cell fluctuations; *(v)* fucose is

mostly constitutive of psychrophiles and halophiles; *(vi)* heavy metal binding is usually co-present with cryoprotective function, suggesting inductive inferencing from co-functionality.

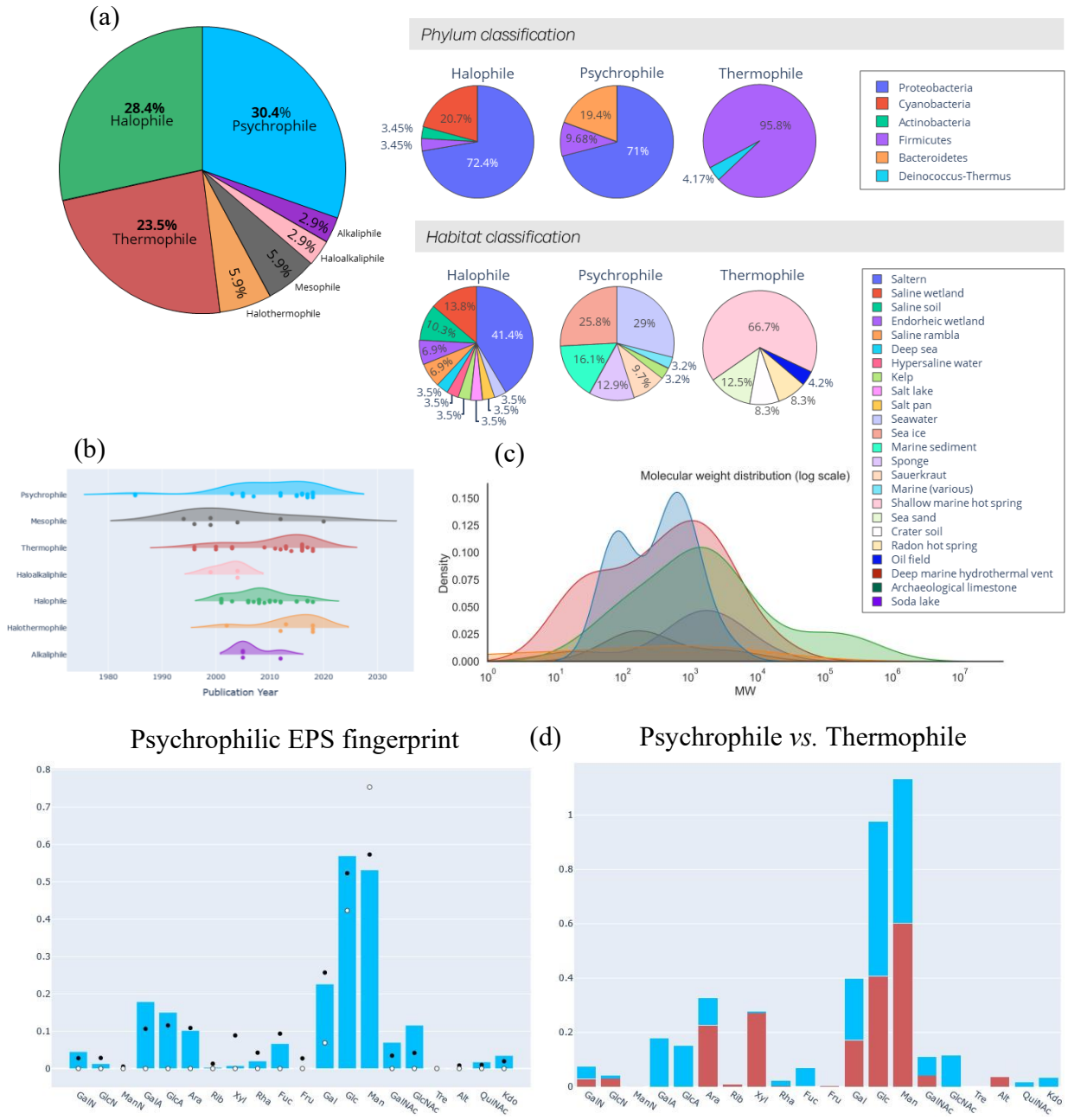


Figure 17. General overview of the extremophilic EPS database. (a) Kingdom and phylum distribution of each producing microorganism, grouped by type of extremophile. (b) Chronological prevalence of extremophile research in the literature, reflecting the waves of interest on habitat-adapted microorganisms. (c) Molecular weight distribution of each EPS by extremophile type, reflecting habitat adaptation. (d) Compositional fingerprinting of each sugar

monomer in psychrophilic (cold) and thermophilic (hot) species, revealing distinct monomer prevalences indicative of specific functionality. *The interactive visualization and respective code can be found here.*

5.1. Dimensionality reduction of relevant variables

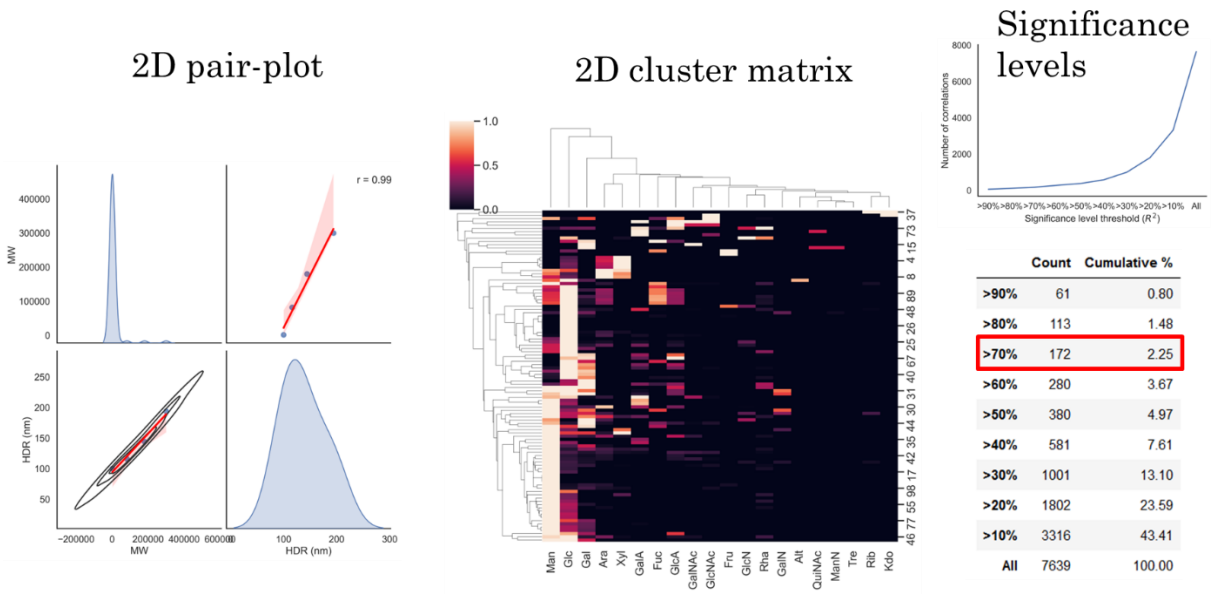
The central purpose of the database was to assess what parameters could be labelled as predictors of cryoprotective function. Adding new entries to the database while characterizing all 144 parameters in the laboratory would be an immense time-consuming and almost impossible task for a PhD, but exploratory data analysis allows to filter which parameters should be focused on in the experimental screening process. However, exploratory analysis is seldom applicable in a raw dataset, and some transformations of the initial variables are required, a method called feature engineering. Techniques of dimensionality reduction are an example of feature engineering.

The first and immediate issue towards establishing a structure-function relational database was the total number of parameters (dimensions) collected. The dimensionality of a database should be drastically smaller than the number of entries (polysaccharides) of which it is composed, at risk of leading to misinterpretation. If the number of dimensions is similar to the number of entries, the issue of infinitesimal dimensionality arises, in which a single value of a parameter might be exclusive to a given entry, not allowing to obtain statistically valid insights. To circumvent this issue and reduce the amount of relevant variables to be considered for analysis, a manual analysis of which parameters should be considered can be carried out, attempting to assess which dimensions are collinear, correlated, and groupable. However, data science tools offer the advantage of time efficiency, automation and high-throughput. Thus, here we describe the initial methodologies employed: pair-plot correlation analysis, one-hot encoding and vector embedding.

5.2. Pair-plot correlation analysis

The first method used was pair-plot correlation analysis (**Figure 18**). Briefly, every single parameter in the database was correlated with all others in a non-discriminatory fashion, which resulted in a total of 7639 correlations. The representation of multiple scatter plots is impractical and consumes critical amounts of computerized processing. Thus, a 2D cluster matrix was used to reflect these correlations, in which lighter colors represent strong correlations between variables. Manually interpreting and selecting over 7000 correlations is unfeasible, and a statistical rationale

was employed. Essentially, the pair correlation factors (R^2) were ordered in a 0–1 normalized scale and grouped by significance thresholds. Any variable pairs showing less than 70% (R^2=0.7) correlation between each other constituted variables for which (i) no strong correlations were found or (ii) not enough data was available. Conversely, variable pairs showing close to or 100% (R^2=1) were immediately discarded, at risk of reflecting unintended collinearity between variables.



	Count	Cumulative %
>90%	61	0.80
>80%	113	1.48
>70%	172	2.25
>60%	280	3.67
>50%	380	4.97
>40%	581	7.61
>30%	1001	13.10
>20%	1802	23.59
>10%	3316	43.41
All	7639	100.00

Figure 18. Summary of exploratory pair-plot correlation analysis. Hundreds of 2D pair plots are expressed as a cluster matrix that simultaneously shows a spectrum of quantitative correlation and inter-relationships between variables. The significance of all variables can be expressed as a cumulative line plot and inform the number of relevant variables to be filtered and analyzed. *The interactive visualization and respective code can be found here.*

From a single iteration of pair-plot analysis variable filtering, a total of 172 correlations remained, which constituted only 2.25% of all possible correlations. From these, parameters like molecular weight, monomer composition, intrinsic viscosity, thermodynamic influence on the freezing point and co-functionalities (heavy metal binding, osmoregulation, antioxidant potential, ice modulation) were pinpointed. From this second iteration, a subset was considered for experimental analysis, which thus far led to critical discoveries of fucose being a major contributor to cryoprotection [17], polysaccharides indulging in ice modulation as a regulator of ice crystal development [21] and their influence in kinetic freezing point change and forming gel states being fundamental towards function [4, *unpublished*].

5.3. One-hot encoding

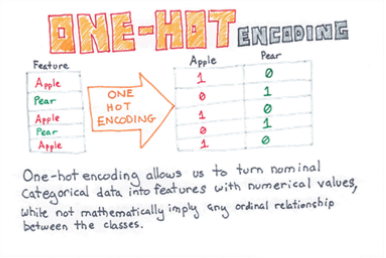
A common-theme issue when dealing with life science datasets in the field of data science is the amount of non-quantitative, categorical data. The extremophilic type of the producing polysaccharide strain (*e.g.* "psychrophile"), the molecular conformation (*e.g.* "alpha-helix", "beta-sheet") and the structural repeating unit (glycosidic linkages, such as α1→6 and β1→4) of a polysaccharide all contain defined text labels which cannot be particularly encoded to a quantifiable or qualifiable value. For these parameters to be fed into Python-based numerical estimators, models and algorithms, the categorical strings must be encoded as numeric features. One-hot encoding allows to map these labels with numerical codes, allowing for them be computed into numerical computations, always enabling a re-conversion back to their original meanings after modelling has been performed. **Figure 19** demonstrates the script and application of this method to transform types of extremophiles and glycosidic linkages into numerical labels.

```
In [6]: # encode the categorical strings as numbers using scikit
        from sklearn.preprocessing import LabelEncoder

        gle = LabelEncoder()
        extremeType_labels = gle.fit_transform(nominalData['ExtremeType'])
        extremeType_mappings = {index: label for index, label in
                                enumerate(gle.classes_)}

        pd.DataFrame.from_dict(extremeType_mappings, orient='index', columns=['ExtremeType'])
```

Out[6]:

	ExtremeType
0	Alkaliphile
1	Haloalkaliphile
2	Halophile
3	Halothermophile
4	Mesophile
5	Psychrophile
6	Thermophile

	Strain	ExtremeType	ExtremeTypeLabel
22	Bacillus thermantarcticus	Thermophile	6
65	Alteromonas macleodii	Mesophile	4
25	Pseudoalteromonas sp. MER144	Psychrophile	5
19	Geobacillus sp. 4004	Thermophile	6
94	Halomonas alkaliantarctica CRSS	Haloalkaliphile	1




```
In [19]: from sklearn.preprocessing import MultiLabelBinarizer

         mlb_linkages = MultiLabelBinarizer()
         onehotLinkages = pd.DataFrame(mlb_linkages.fit_transform(onehotData['LinkageTypes'].str.split('; ')),
                                       columns=mlb_linkages.classes_,
                                       index=onehotData['LinkageTypes'].index)

         onehotLinkages = onehotLinkages.drop('', 1)
         onehotLinkages
```

Out[19]:

	?(1→2)	?(1→3)	?(1→4)	?(1→5)	?(1→6)	α(1→2)	α(1→3)	α(1→4)	α(1→6)	α(2→2)	β(1→2)	β(1→3)	β(1→4)	β(1→6)	β(2→6)
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
99	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
100	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0

102 rows × 15 columns

Figure 19. Overview of one-hot encoding theory and application to the extremophilic database.

All previous visualizations in **Figures 17** and **18** benefitted from the consistent usage of one-hot encoding. The encoding of glycosidic linkages in the structural repeating unit is an exceptional example, whereas instead of finite numeral labelling, a binary transformation (0 or 1) is used (**Figure 19**). In this case, binary encoding allows to expand the database into a matrix, whereas all possible values of a dimension (*e.g.* "α1→6" as value of the dimension "Linkages") becomes a dimension itself. In this fashion, the structural repeating unit of a polysaccharide can be discretized into a matrix, which can reveal trends in outcome. For instance:

1. α-linkages (1→2 or 1→6) result in structure flexibility (*e.g.* dextrans);
2. β-linkages (1→4 or 1→3) result in structure rigidity (*e.g.* cellulose and the cellulosic backbone of xanthan);
3. α-glucans (e.g. dextran) are water-soluble, while β-glucans (e.g. cellulose, curdlan) are water-insoluble. A β-glucan is only soluble when it contains more than one β(1→4) linkage (*e.g.* alginate).
4. β-linkages that are not 1→4 but constituted of glucose are water-soluble (*e.g.* β(1→3) of laminarin). However curdlan is not soluble despite having a β(1→3) backbone because it does not contain branched β(1→6) linkages.
5. β-linkages that are not 1→4 but not constituted of glucose, like the β(1→2) sucrose linkages of Ficoll™, are water-soluble.

The drawback of one hot-encoding is that, contrary to dimensionality reduction, it increases the number of dimensions in the database, for which vector embedding can provide a solution.

5.1. Vector embedding

Vector embedding is the opposite method to one-hot encoding. Here, an application example will be given for the monomer composition of each polysaccharide, but the same method can be used for the previous problem. Briefly, vector embedding is the engineering of 1-dimensional vectors which capture the information contained in *n*-dimensional parameters. A proper mathematical analogy is a 3×3 matrix, which captures the information of 9 parameters (each value in the matrix) in a single object. From a programming perspective, the matrix is a single 1-dimensional object.

In this database, the monomer composition of each polysaccharide is reported for 25 different monomers, as a normalized molar ratio from 0 to 1. One can easily deduce that the prevalence of,

for example, glucose and rhamnose, are mostly independent variables, and the pair correlation between them does not constitute verifiable nor a physically valid form of determining cryoprotective function. However, a 1D vector that contains quantitative information of 25 dimensions provides a comparable object between different polysaccharides. In essence, such a vector acts as a compositional fingerprint for that polysaccharide and can be represented in 25-D space. Likewise, the average molar ratio of each monomer from every extremophilic type can also be embedded in an average vector that represents a specific class, *e.g.* psychrophile vector, thermophile vector.

Figure 20 represents the compositional fingerprinting for different extremophilic types, before vectorization. Galactose (Gal), glucose (Glc) and mannose (Man) are ubiquitously present monomers amongst all classes, indicating the absence of bioactive relevance with respect to no particular functionality. However, valuable insights can be drawn from the comparative fingerprint analysis. **Figure20*b*** reveals that there is a high similarity between psychrophilic and halophilic compositions, namely in their uronic acids, fucose (and rhamnose as chemical equivalent) contents, which is significant distinction from thermophilic compositions. This suggests two things: (i) that these monomers are biologically relevant, by a natural selection perspective, towards dealing with the external stressors that arise in cold and highly saline habitats, that are not present in high temperature environments; and (ii) that the high similarity between psychrophiles and halophiles may suggest not only that halophiles may be cryoprotectant, but that their identifying stressors may arise from a more fundamental and common denominator, which appears to be cell volume fluctuations during to osmotic flow in an attempt to preserve cellular homeostasis.

In conclusion, vector embedding of monomer compositions (which essentially is a 1×25 matrix) cannot be represented visually, but can be computed for similarities between vectors, revealing similarities and distinctions between extremophilic types, which may reflect how specific combinations of monomers results in environmental adaptation to its characteristic external stress. Further modelling of the fingerprints shown in **Figure 20** can be implemented in visual *t*-SNE methodologies (*t*-distributed Stochastic Neighbor Embedding) which allow to visualize high-dimensional data in clusters with similar probability distributions, according to a Kullback-Leibler divergence metric. The latter analysis is currently under way.

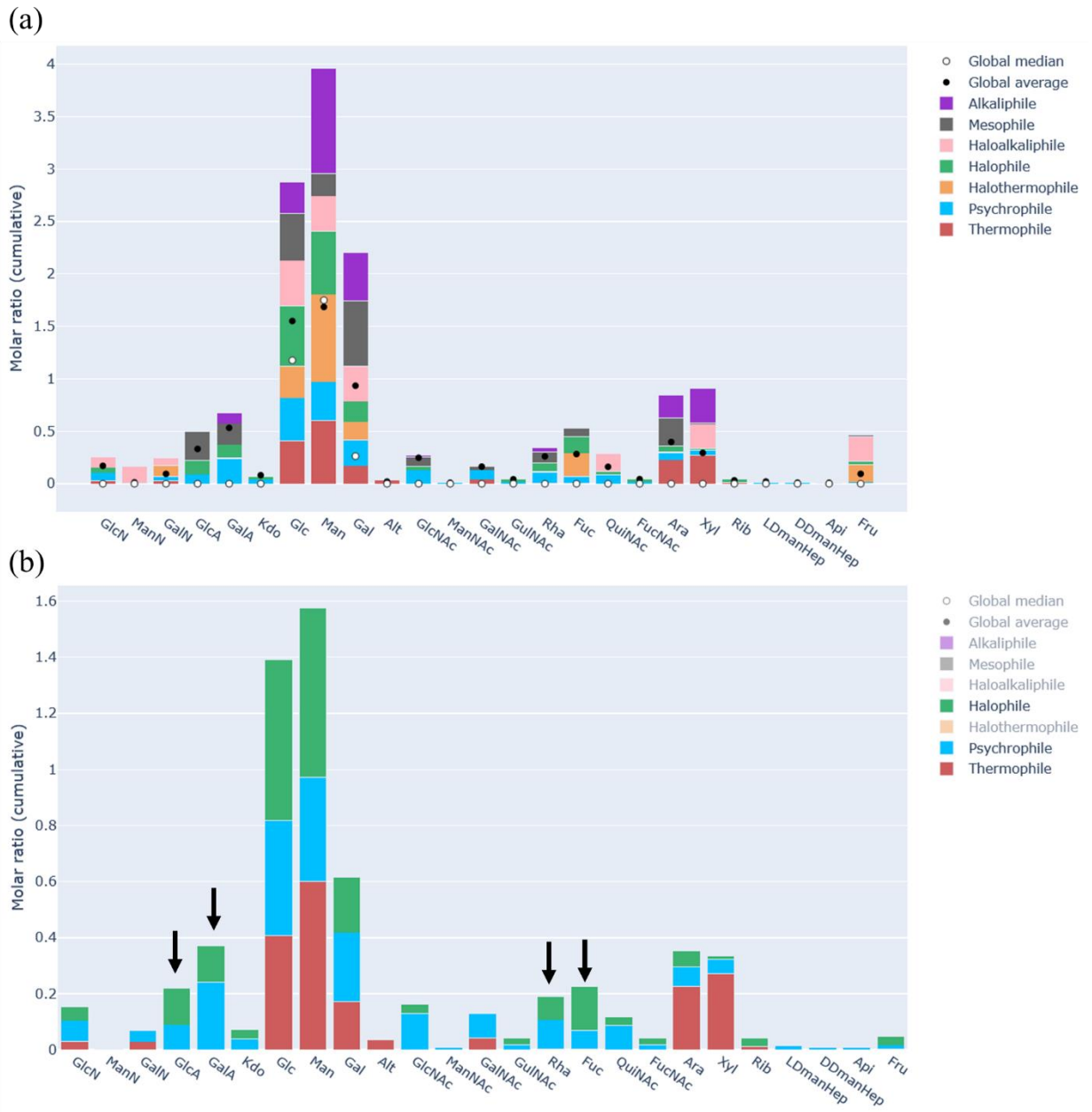


Figure 20. Average compositional fingerprint of each extremophilic type. Panel (a) shows the general overview of all fingerprints for every single class. Notice that galactose (Gal), glucose (Glc) and mannose (Man) are ubiquitously present amongst all classes. Panel (b) shows a comparison between heat-adapted thermophiles (red), cold-adapted psychrophiles (blue) and high salinity-adapted halophiles (green). The high similarity between psychrophiles and halophiles might suggest a common-denominator functional protection against external stresses. *The interactive visualization and respective code can be found here.*

5.2. How did using Data Science solve this problem?

The results from exploratory data analysis and feature engineering carried out in this database were the major driving force for filtering the experimental workload performed during the PhD thesis, which truthfully allowed to discern the most crucial properties to be assessed, in order to predict a cryoprotective outcome. The initial stage of the PhD had an impromptu shift in focus due to the onset of a pandemic which did not allow for experimental data to be collected. Therefore, acquiring vast expertise in multidimensional analysis and generating a database of this size in order to reveal previously hidden structure-function trends was an adaptation towards the conditions provided at the time.

Conclusion

The self-learning journey into programming and data science, which arose from necessity, led to fortunate knowledge in the data science field and allowed for life science problems to be solved. Although the most primitive applications of these tools are shown in application #4, the culmination of all that knowledge culminated in application #1, where expertise out of the field of Biochemistry was shown in Berkeley, for such work could not have been performed with pre-PhD skills. In total, an approximate minimum of 350 hours of learning were carried out in online supervised specialization in the year 2020, with an additional 150 hours involving capstone projects for the conclusion of those specializations. These capstone projects involved working with Python and R development, SQL database queries, computer vision algorithms, coding from scratch machine learning algorithms and neural networks, both for image recognition and natural language processing, data science capstone projects involving the analysis of large datasets in the life sciences, atmospheric, entertainment and financial fields, and the development of various interactive web applications, couple to advanced mathematical and statistical tools. Although not all knowledge was duly implemented in the PhD thesis, the continuous practice of a different expertise consolidated a problem-solving capacity that differs from how problems are solved in life sciences and broadened my critical thinking capabilities and efficiency.

Part of the research documentation presented herein,
which constitutes several papers currently under review,
was awarded with a ***Student Best Poster Presentation*** *prize*
where the ***PCA method*** *highlighted the importance of fucose in cryopreservation,*
at the 11th European Symposium on Biopolymers, Brno, Czech Republic (2023).

References

[1] M. Münch, C. Raab, M. Biehl, and F. M. Schleif, "Data-Driven Supervised Learning for Life Science Data," *Front. Appl. Math. Stat.*, vol. 6, Nov. 2020, doi: 10.3389/FAMS.2020.553000/EPUB.

[2] A. Dudek, T. Arodz, and J. Galvez, "Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review," *Comb. Chem. High Throughput Screen.*, vol. 9, no. 3, pp. 213–228, Feb. 2006, doi: 10.2174/138620706776055539.

[3] A. N. Consiglio, D. Lilley, R. Prasher, B. Rubinsky, and M. J. Powell-Palm, "Methods to stabilize aqueous supercooling identified by use of an isochoric nucleation detection (INDe) device," *Cryobiology*, Mar. 2022, doi: 10.1016/J.CRYOBIOL.2022.03.003.

[4] B. M. Guerreiro, A. N. Consiglio, B. Rubinsky, M. J. Powell-Palm, and F. Freitas, "Enhanced Control over Ice Nucleation Stochasticity Using a Carbohydrate Polymer Cryoprotectant," *ACS Biomater. Sci. Eng.*, p. acsbiomaterials.2c00075, Apr. 2022, doi: 10.1021/ACSBIOMATERIALS.2C00075.

[5] D. Lilley, J. Lau, C. Dames, S. Kaur, and R. Prasher, "Impact of size and thermal gradient on supercooling of phase change materials for thermal energy storage," *Appl. Energy*, vol. 290, p. 116635, May 2021.

[6] B. M. Guerreiro, F. Freitas, J. C. Lima, J. C. Silva, M. Dionísio, and M. A. M. Reis, "Demonstration of the cryoprotective properties of the fucose-containing polysaccharide FucoPol," *Carbohydr. Polym.*, vol. 245, p. 116500, Oct. 2020, doi: 10.1016/J.CARBPOL.2020.116500.

[7] E. K. Bigg, "The supercooling of water," *Proc. Phys. Soc. Sect. B*, vol. 66, no. 8, pp. 688–694, 1953, doi: 10.1088/0370-1301/66/8/309.

[8] K. A. Murray and M. I. Gibson, "Chemical approaches to cryopreservation," *Nat. Rev. Chem. 2022 68*, vol. 6, no. 8, pp. 579–593, Jul. 2022, doi: 10.1038/s41570-022-00407-4.

[9] B. Kirchner and M. Reiher, "The secret of dimethyl sulfoxide-water mixtures. A quantum chemical study of 1DMSO-nwater clusters," *J. Am. Chem. Soc.*, vol. 124, no. 21, pp. 6206–6215, May 2002, doi: 10.1021/JA017703G.

[10] C. Polge, A. Smith, and A. Parkes, "Revival of spermatozoa after vitrification and dehydration at low temperatures.," *Nature*, vol. 164, no. 4172, p. 666, Oct. 1949, Accessed: Sep. 23, 2018. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18143360.

[11] K. W. Yong, L. Laouar, J. A. W. Elliott, and N. M. Jomha, "Review of non-permeating cryoprotectants as supplements for vitrification of mammalian tissues," *Cryobiology*, vol. 96, pp. 1–11, Oct. 2020, doi: 10.1016/J.CRYOBIOL.2020.08.012.

[12] C. B. Marshall, G. L. Fletcher, and P. L. Davies, "Hyperactive antifreeze protein in a fish," *Nature*, vol. 429, no. 6988, pp. 153–153, May 2004, doi: 10.1038/429153a.

[13] Y. Yeh and R. E. Feeney, "Antifreeze proteins: Structures and mechanisms of function," *Chem. Rev.*, vol. 96, no. 2, pp. 601–618, 1996, doi: 10.1021/CR950260C/ASSET/IMAGES/LARGE/CR950260CF00009.JPEG.

[14] A. T. Rahman *et al.*, "Ice recrystallization is strongly inhibited when antifreeze proteins bind to

multiple ice planes," *Sci. Reports 2019 91*, vol. 9, no. 1, pp. 1–9, Feb. 2019, doi: 10.1038/s41598-018-36546-2.

[15] C. J. Capicciotti, J. D. R. Kurach, T. R. Turner, R. S. Mancini, J. P. Acker, and R. N. Ben, "Small Molecule Ice Recrystallization Inhibitors Enable Freezing of Human Red Blood Cells with Reduced Glycerol Concentrations," *Sci. Reports 2015 51*, vol. 5, no. 1, pp. 1–10, Apr. 2015, doi: 10.1038/srep09692.

[16] C. J. Capicciotti *et al.*, "Potent inhibition of ice recrystallization by low molecular weight carbohydrate-based surfactants and hydrogelators," *Chem. Sci.*, vol. 3, no. 5, pp. 1408–1416, Apr. 2012, doi: 10.1039/C2SC00885H.

[17] B. M. Guerreiro *et al.*, "Fucose is an essential feature in cryoprotective polysaccharides," *bioRxiv*, p. 2023.10.13.562212, Oct. 2023, doi: 10.1101/2023.10.13.562212.

[18] M. Schneider, E. Al-Shareffi, and R. S. Haltiwanger, "Biological functions of fucose in mammals," *Glycobiology*, vol. 27, no. 7, p. 601, Jul. 2017, doi: 10.1093/GLYCOB/CWX034.

[19] C. Stoll and W. F. Wolkers, "Membrane Stability during Biopreservation of Blood Cells," *Transfus. Med. Hemotherapy*, vol. 38, no. 2, p. 89, Apr. 2011, doi: 10.1159/000326900.

[20] A. K. Balcerzak, M. Febbraro, and R. N. Ben, "The importance of hydrophobic moieties in ice recrystallization inhibitors," *RSC Adv.*, vol. 3, no. 10, pp. 3232–3236, Feb. 2013, doi: 10.1039/C3RA23220D.

[21] B. Guerreiro, L. T. Lou, B. Rubinsky, and F. Freitas, "Ice modulatory effect of the polysaccharide FucoPol in directional freezing," *Soft Matter*, 2023, doi: 10.1039/D3SM01154B.