Retrieval algorithm for JWST Infrared ice spectra using ML and MCMC

Pabitra Ranjan Giri, 2011103
Computational Physics(P452) Term Paper
April 24, 2024
School of Physical Sciences, NISER, HBNI, India

JWST ice spectra contains infrared absorption features. In this project we use Machine learning and Markov Chain Monte Carlo techniques to retrieve the component spectra of an observed spectrum. The spectral analysis include, continuum removal, silicate removal, and retreival. This project only focuses on retrieval algorithm.

I. THEORY

A. Infrared ice spectrum analysis

Infrared (IR) observations of ice using the James Webb Space Telescope (JWST) involve several key theoretical concepts and equations.

1. Blackbody Radiation:
- Blackbody radiation describes the electromagnetic radiation emitted by a perfect absorber and emitter of radiation at all wavelengths and temperatures. The intensity of blackbody radiation emitted by an object depends on its temperature and wavelength. - The spectral radiance (B_λ) of a blackbody at a given wavelength (λ) and temperature (T) is given by Planck's law:

$$B_\lambda(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1}$$

where: - h is Planck's constant, - c is the speed of light, - k is Boltzmann's constant.

2. Emission Spectrum of Ice and Analysis:
Ice emits infrared radiation due to its temperature according to its emission spectrum, which is dependent on its temperature and the physical properties of ice. Ice absorption features in the infrared region are primarily due to molecular vibrations of water molecules. These features are distinctive and can be used to identify the presence of ice and infer its physical properties. Ice observation in the infrared involves analyzing the absorption and emission features of ice in the infrared spectrum to characterize its properties such as composition, temperature, and abundance.

3. Data Analysis:
Data obtained from JWST observations are analyzed using computational techniques such as spectral fitting, radiative transfer modeling, and comparison with theoretical spectra. Comparison with theoretical spectra generated from laboratory experiments and theoretical models helps validate observations and refine our understanding of ice in different astrophysical contexts.

B. Markov Chain Monte Carlo(Metropolis Hastings)

Markov Chain Monte Carlo (MCMC) methods, particularly the Metropolis-Hastings algorithm, are powerful tools for sampling from complex probability distributions. Here's a theoretical explanation along with the key formulas:

1. Markov Chains:
- A Markov chain is a stochastic process where the future state depends only on the current state and not on the sequence of events that preceded it. A Markov chain satisfies the Markov property:

$$P(X_{t+1}|X_0, X_1, ..., X_t) = P(X_{t+1}|X_t)$$

- The state space of a Markov chain consists of all possible states that the system can occupy, and the transition probabilities between states are typically represented by a transition matrix.

2. Monte Carlo Methods:
- Monte Carlo methods are computational techniques that use random sampling to estimate numerical results. In the context of MCMC, Monte Carlo methods are employed to sample from complex probability distributions.

3. Metropolis-Hastings Algorithm:
- The Metropolis-Hastings algorithm is a MCMC method used to sample from a target probability distribution $\pi(x)$ when direct sampling is difficult.
- Given the current state $x^{(t)}$, the algorithm proposes a new state x' based on a proposal distribution $q(x'|x^{(t)})$.
- The acceptance probability $\alpha(x^{(t)}, x')$ is calculated as the ratio of the target distribution at the proposed state to the target distribution at the current state, multiplied by the ratio of the proposal densities:

$$\alpha(x^{(t)}, x') = \min\left(1, \frac{\pi(x')}{\pi(x^{(t)})} \cdot \frac{q(x^{(t)}|x')}{q(x'|x^{(t)})}\right)$$

- If $\alpha(x^{(t)}, x')$ is greater than or equal to a randomly generated value u from a uniform distribution between 0 and 1, the proposal x' is accepted, and the chain moves to the new state. Otherwise, the current state is retained.

4. Detailed Balance:
- The Metropolis-Hastings algorithm satisfies the detailed balance condition, which ensures that the stationary distribution of the Markov chain is the target distribution $\pi(x)$. Mathematically, this condition can be expressed as:

$$\pi(x^{(t)}) \cdot T(x^{(t)}, x') = \pi(x') \cdot T(x', x^{(t)})$$

where $T(x^{(t)}, x')$ is the transition probability from state $x^{(t)}$ to state x'.

5. Convergence and Burn-in:
- MCMC algorithms such as Metropolis-Hastings may require an initial "burn-in" period where the chain converges to the target distribution. Subsequent samples are then used for inference.

C. Machine learning algorithm: Non-Negative Least square

The Non-Negative Least Squares (NNLS) algorithm is a method used for solving linear least squares problems with non-negativity constraints on the coefficients.

1. Linear Least Squares Problem:
- Given a matrix X of size $m \times n$ and a vector y of size $m \times 1$, the linear least squares problem aims to find a vector β of size $n \times 1$ that minimizes the squared Euclidean norm of the residual vector $y - X\beta$:

$$\min_{\beta} \|y - X\beta\|^2$$

2. Non-Negative Least Squares Problem:
- In some applications, it is desirable to enforce non-negativity constraints on the coefficients of β. The non-negative least squares problem is formulated as follows:

$$\min_{\beta \geq 0} \|y - X\beta\|^2$$

3. Non-Negative Least Squares Algorithm:
- The NNLS algorithm solves the non-negative least squares problem iteratively using methods such as the active set strategy or projected gradient descent.
- One common approach is the active set strategy, which starts with an initial solution and iteratively updates the active set of variables (i.e., variables with non-zero coefficients) until convergence.

4. Active Set Strategy:
- At each iteration, the active set A is updated based on the current solution β by identifying the indices of coefficients that are not at their lower bounds (i.e., non-zero).
- The algorithm then solves a reduced least squares problem using only the variables in the active set, with non-negativity constraints enforced.
- If the solution to the reduced problem contains non-negative coefficients, it is accepted as the new solution. Otherwise, the active set is updated to remove variables with negative coefficients, and the process repeats until convergence.

5. Convergence:
- Convergence of the NNLS algorithm is typically assessed based on the change in the objective function value or the norm of the gradient between iterations.
- The algorithm terminates when the change in the objective function value is below a specified tolerance or when a maximum number of iterations is reached.

6. Mathematical Formulation:
- Let X be the $m \times n$ matrix of predictors, y be the $m \times 1$ vector of responses, and β be the $n \times 1$ vector of coefficients.
- The non-negative least squares problem can be formulated as the following optimization problem:

$$\min_{\beta \geq 0} \|y - X\beta\|^2$$

For this project, I have used active set method.

II. ALGORITHM

A. Interpolation

The individual spectra used for analysis have data-points at different 'X' values. To fix this we use cubic interpolation to interpolate the spectra in our region of interest. I have used "interp1d" function from "scipy.interpolate" library file to perform the interpolation. Also additional snippets to add zeros where, the spectra doe not have any nearby data-points.

B. Synthetic spectra generator(Absorbance and flux units)

After interpolation of the individual spectra, now we can perform analysis. First we need a synthetic spectrum generator, which can combine different molecular spectra to form a synthetic spectra. First we need to re-evaluate the absorbances(Y) for each spectra according to rquired column density. Each laboratory spectra is observed for a given column density. The expression for new column density is,

$$A^{req} = \frac{N^{req}}{N^{lab}} A^{lab}$$

where,
A^{req} = required absorbance values
A^{lab} = laboratory absorbance values
N^{req} = required column density

N^{lab} = laboratory column density

Now, combining these spectras is just a linear combination.

$$A^{syn} = \Sigma_i w_i A_i^{lab}$$

where, $w_i = \frac{N_i^{req}}{N_i^{lab}}$

The synthetic spectra generator will be used in generating spectras, after MCMC coefficient sampling.

For generating the spectra in flux units, we use the formula,

$$F_{syn} = F_{continuum} e^{-A_{syn}}$$

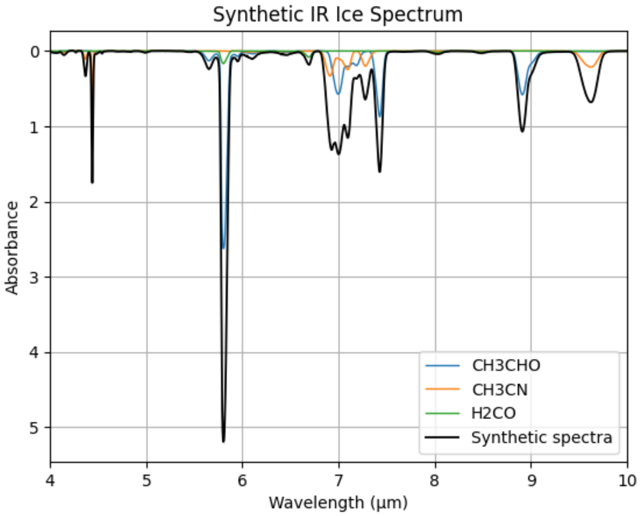

FIG. 1: Synthetic spectra(absorbance units) for molecular components

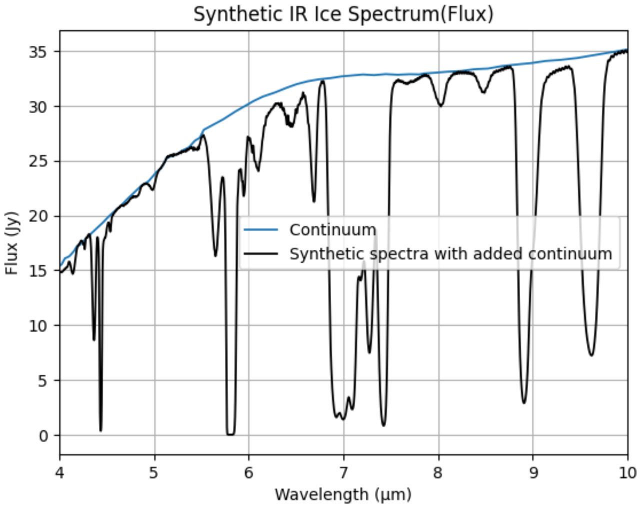

FIG. 2: Synthetic spectrum in Flux units

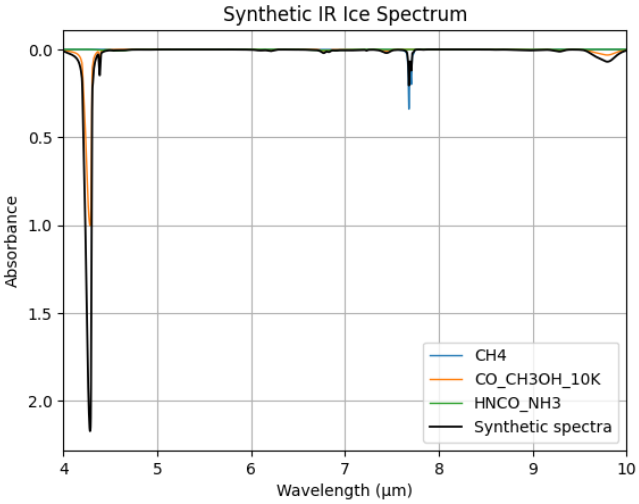

FIG. 3: Synthetic spectra(absorbance units) for molecular components

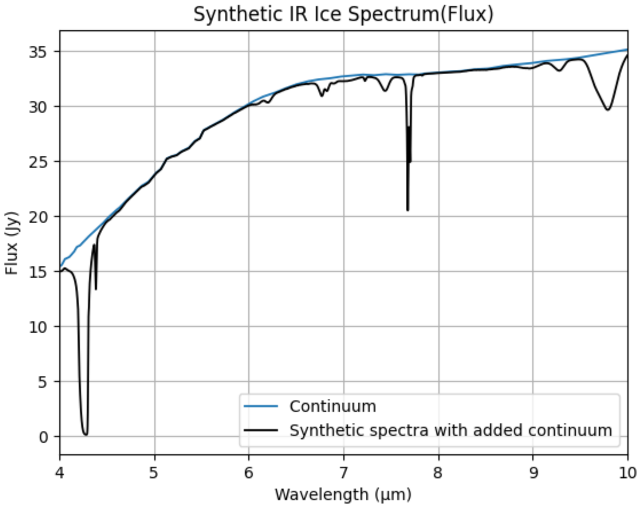

FIG. 4: Synthetic spectrum in Flux units

C. Fitting using ML

For ML I use a Non-Negative least square algorithm because the weights/coefficients of each of the spectra cannot be negative.

My 'X' data consists of the absorbance values for all the lab spectra. Suppose , I have 10 lab spectra with 1000 datapoints each. Then my 'X' data is 1000 x 10 matrix.

The 'Y' data consists of the absorbance values of the JWST spectrum data.

Now, I train the model with 80% of the dataset and test with 20% data. This gives a very accurate fit to the actual spectrum. There may be a possibilty of overfitting.

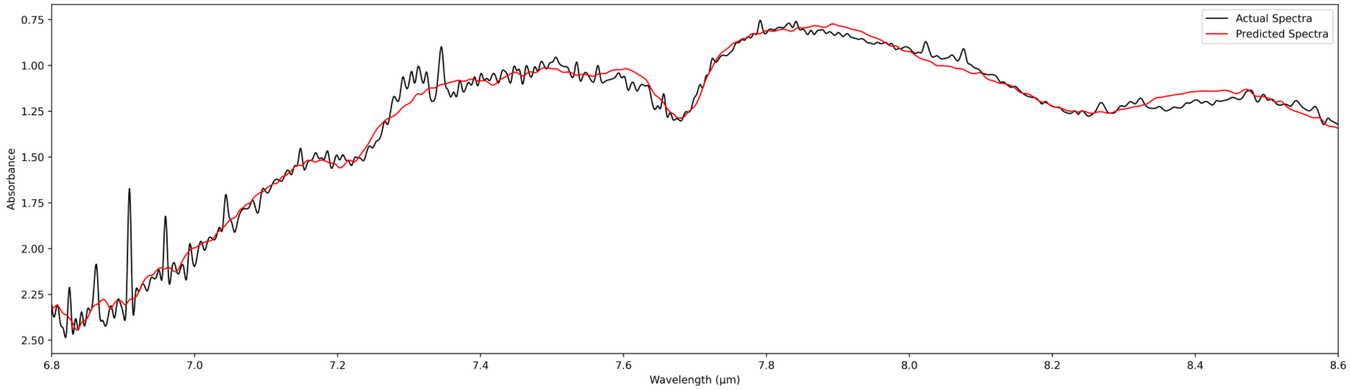

FIG. 5: ML fit of the JWST spectrum

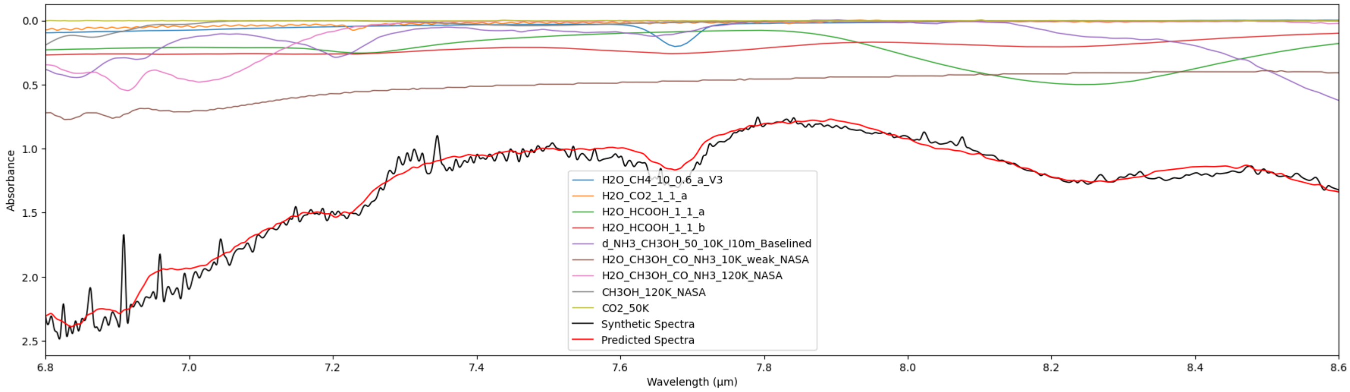

FIG. 6: Component spectra as calculated from ML fit

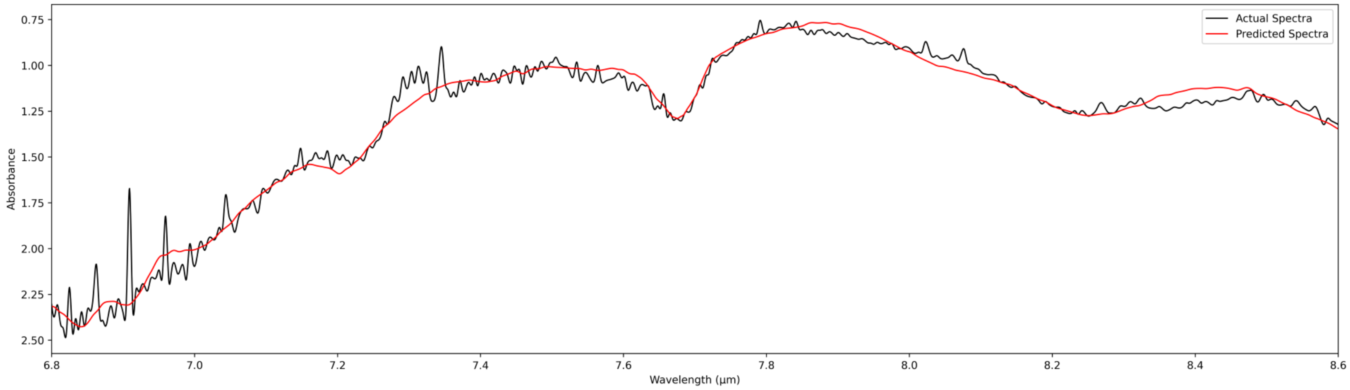

FIG. 7: ML fit using filtered components(correlation>0.5)

D. Fitting using MCMC

To avoid overfitting we select molecular spectra with correlation greater than 0.5 . Their coefficient/weights are the sampling variables for the Metropolis hastings algorithm. Which then samples the weights to find the best fit.

As, we increase the number of sampling the predicted spectra becomes more accurate, but computation time also increases, so we do not use this to completely fit. But only as a re-checker that the combination of the filtered spectra tends to actual spectra at higher number of samplings.

E. Filtering Components

Now, we filter the components based on the coefficients given by MCMC algorithm. We only use components with a significant coefficient(>0.1) . Now, the feed these selected spectra into the ML algorithm once again.

The 'X' data now consists of the selected spectra only.

This gives a more accurate fit to our JWST spectra without any overfitting.

F. Final statistics

Now, we can use this spectra to find the mean-squared error of the second ML fit and weights of the coefficients and generate the predicted spectra.

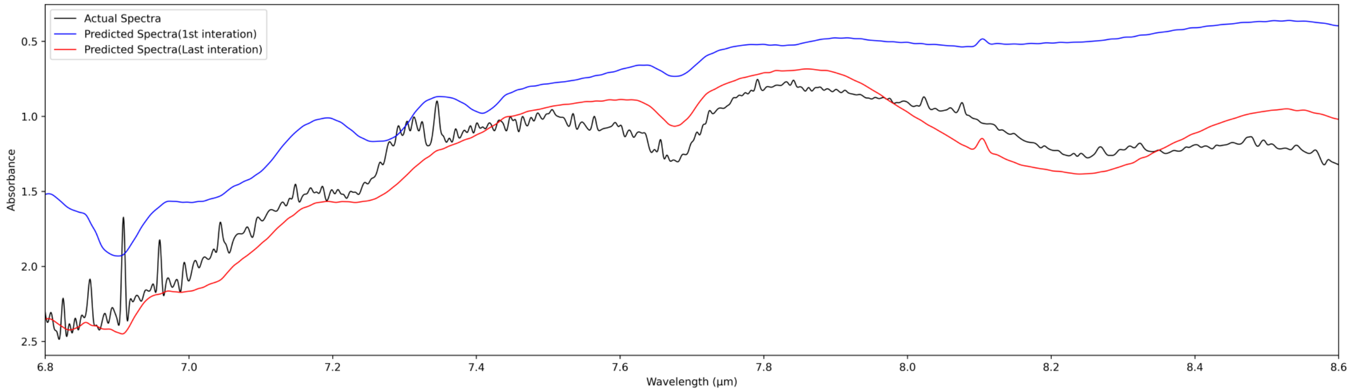

FIG. 8: MCMC fit using filtered components(correlation>0.5)

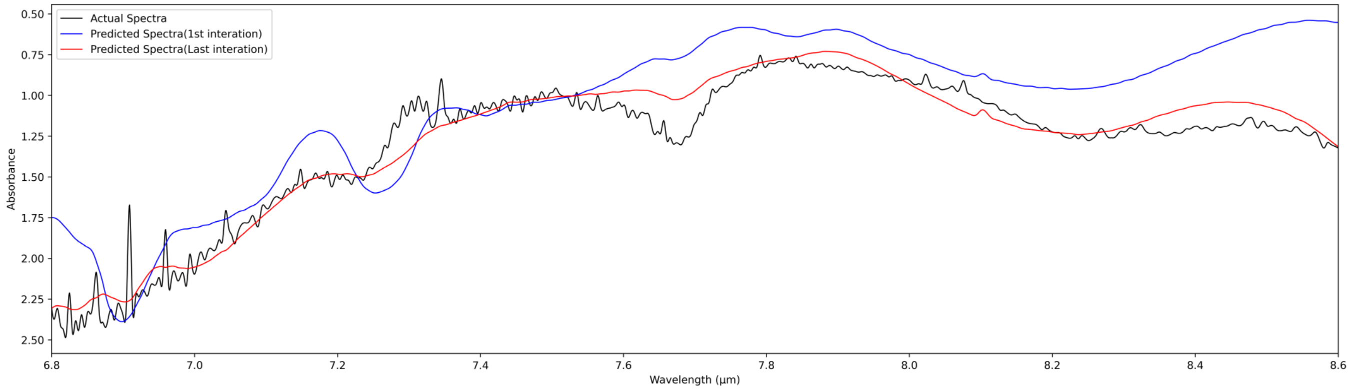

FIG. 9: MCMC fit(100000 samples) using filtered components(correlation>0.5)

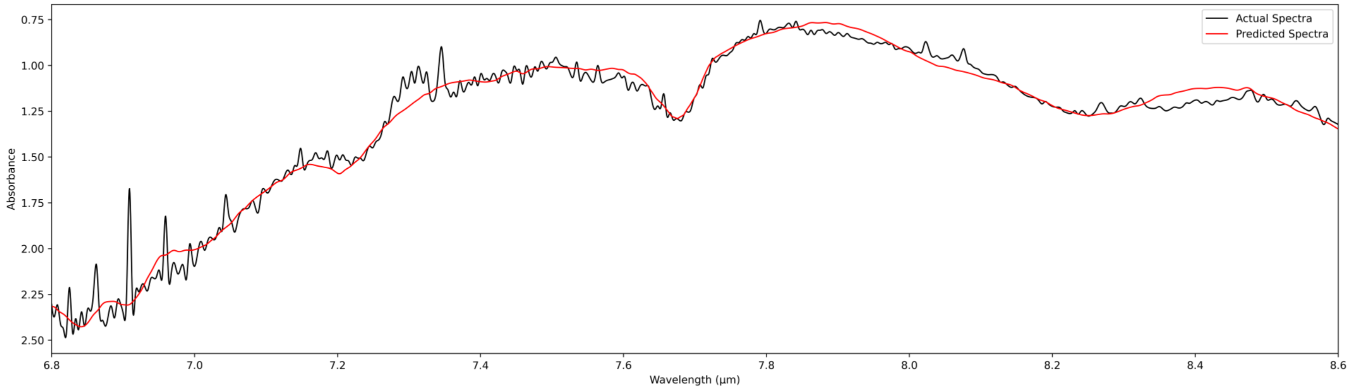

FIG. 10: Second ML fit using filtered MCMC components(coefficient>0.1)

III. RESULTS AND CONCLUSION

- All the analysis was done for the complete spectra(2-20 μm), but results are shown only for region(6.8-8.6 μm) so that the result can be compared to a research paper.

- The first ML fit has a mean-squared error = 0.0039051

- The second ML fit has a mean-squared error = 0.0045153

- The Metropolis Hastings has a mean-squared error for = 0.14997

- The research states the major components of the JWST spectra to be : CH_4, $HCOOH$, CH_3CHO, CH_3CH_2OH, CH_3COOH, CH_3OCHO.

- The algorithm prepared in this project predicts the major components to be : CH_4, $HCOOH$, CH_3CHO, CH_3CH_2OH, NH_3.

- The algorithm predicts half of the major components accurately.

IV. REFERENCES

- JWST Observations of Young protoStars (JOYS+): Detecting icy complex organic molecules and ions

- Fitting infrared ice spectra with genetic modelling algorithms

- LIDA: The Leiden Ice Database for Astrochemistry

- Metropolis–Hastings algorithm

- Non-negative least squares for high-dimensional linear models:consistency and sparse recovery without regularization