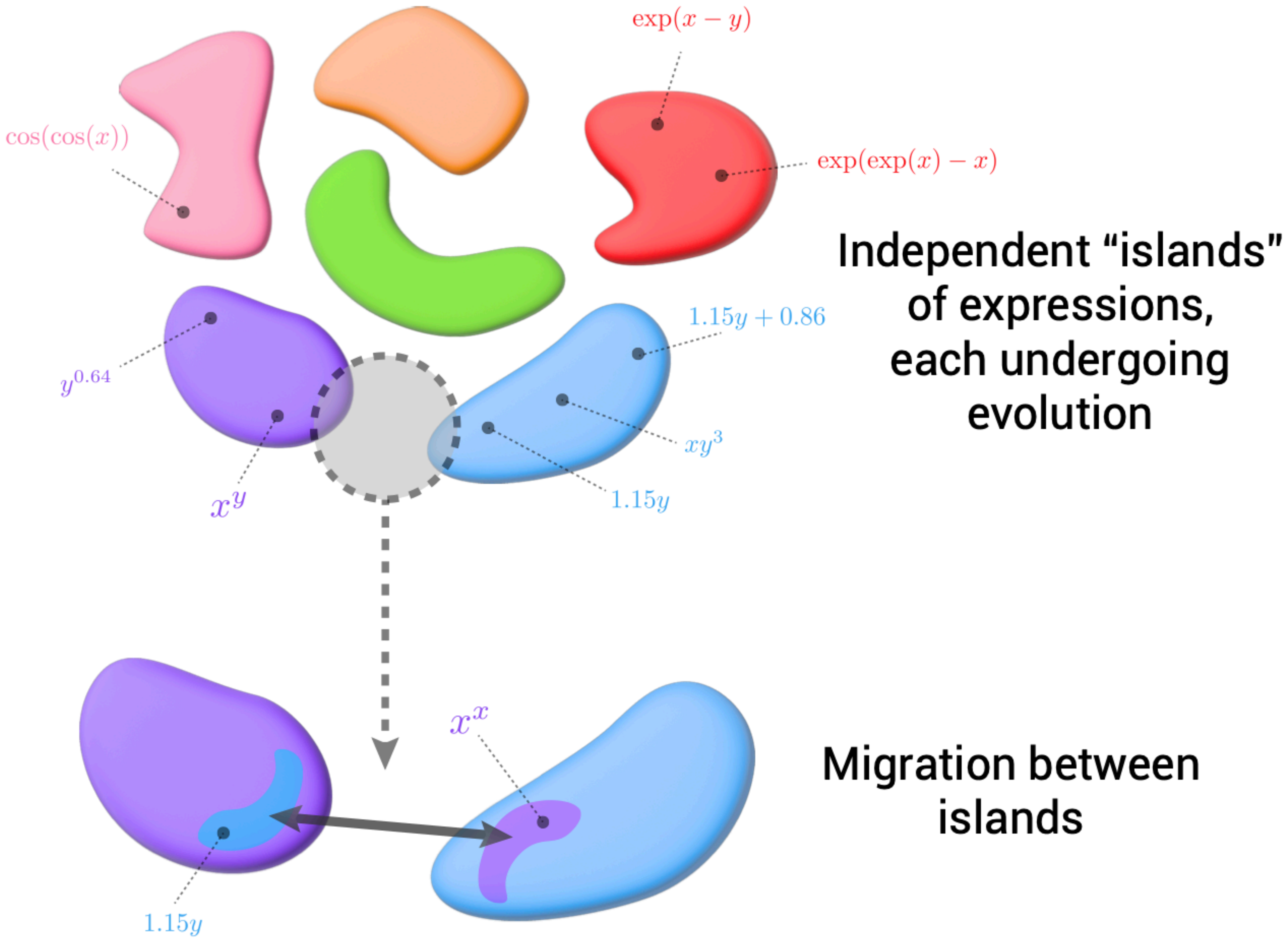
Using Symbolic Regression to Analyse Plasma Data

How PySR works:

PySR is a Symbolic Regression that uses any unary or binary operators to return the best fit it finds with those operators.

PySR is an evolutionary algorithm trained with “natural” selection. The different populations can also interact with exchanges of branches.

The algorithm, then, chooses the equation based on a the log-loss complexity curve, meaning a lower complexity may be chosen even with a higher loss.



Independent “islands” of expressions, each undergoing evolution

Migration between islands

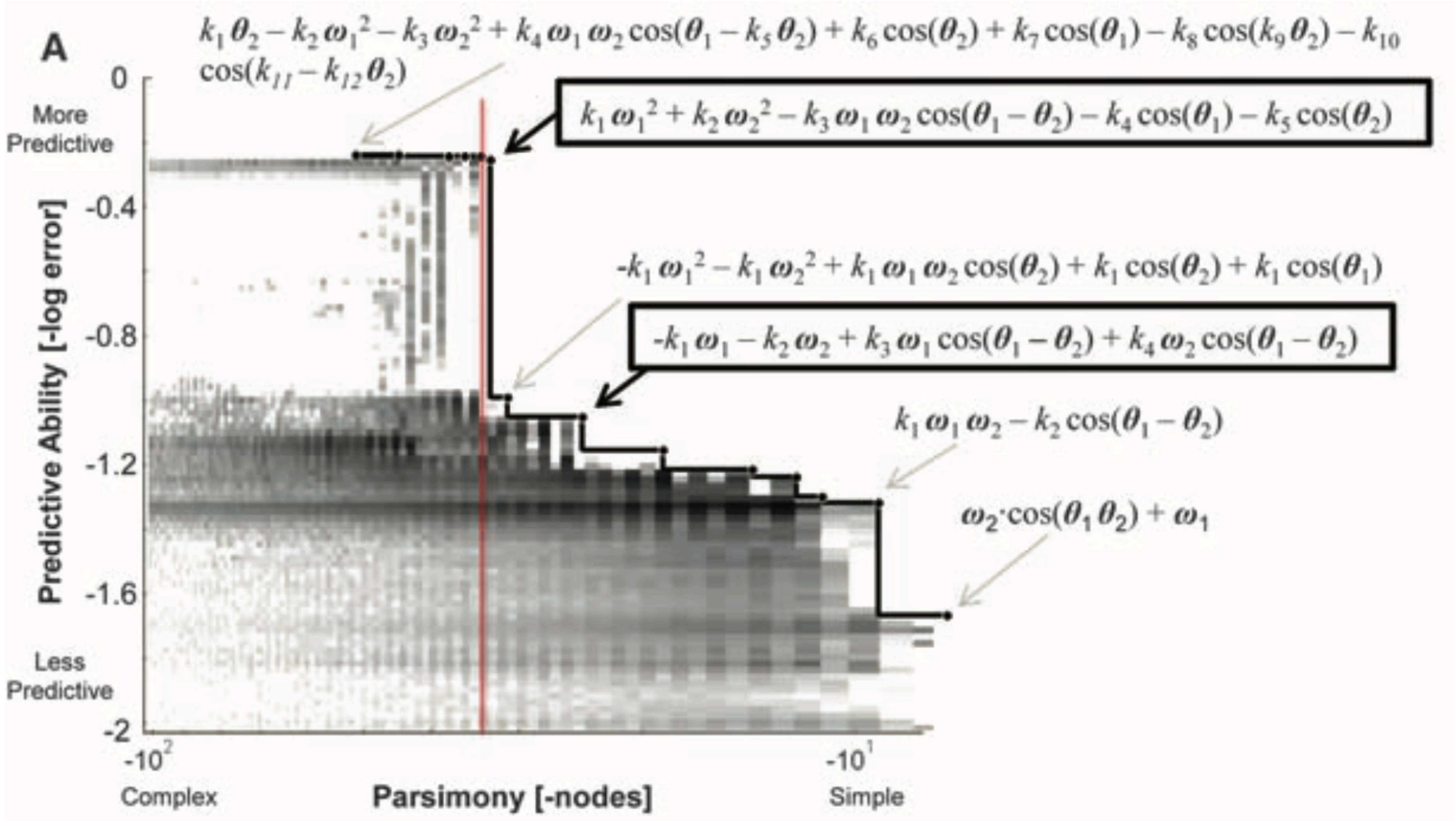
Using Symbolic Regression to Analyse Plasma Data

How PySR works:

PySR is a Symbolic Regression that uses any unary or binary operators to return the best fit it finds with those operators.

PySR is an evolutionary algorithm trained with “natural” selection. The different populations can also interact with exchanges of branches.

The algorithm, then, chooses the equation based on a the log-loss complexity curve, meaning a lower complexity may be chosen even with a higher loss.



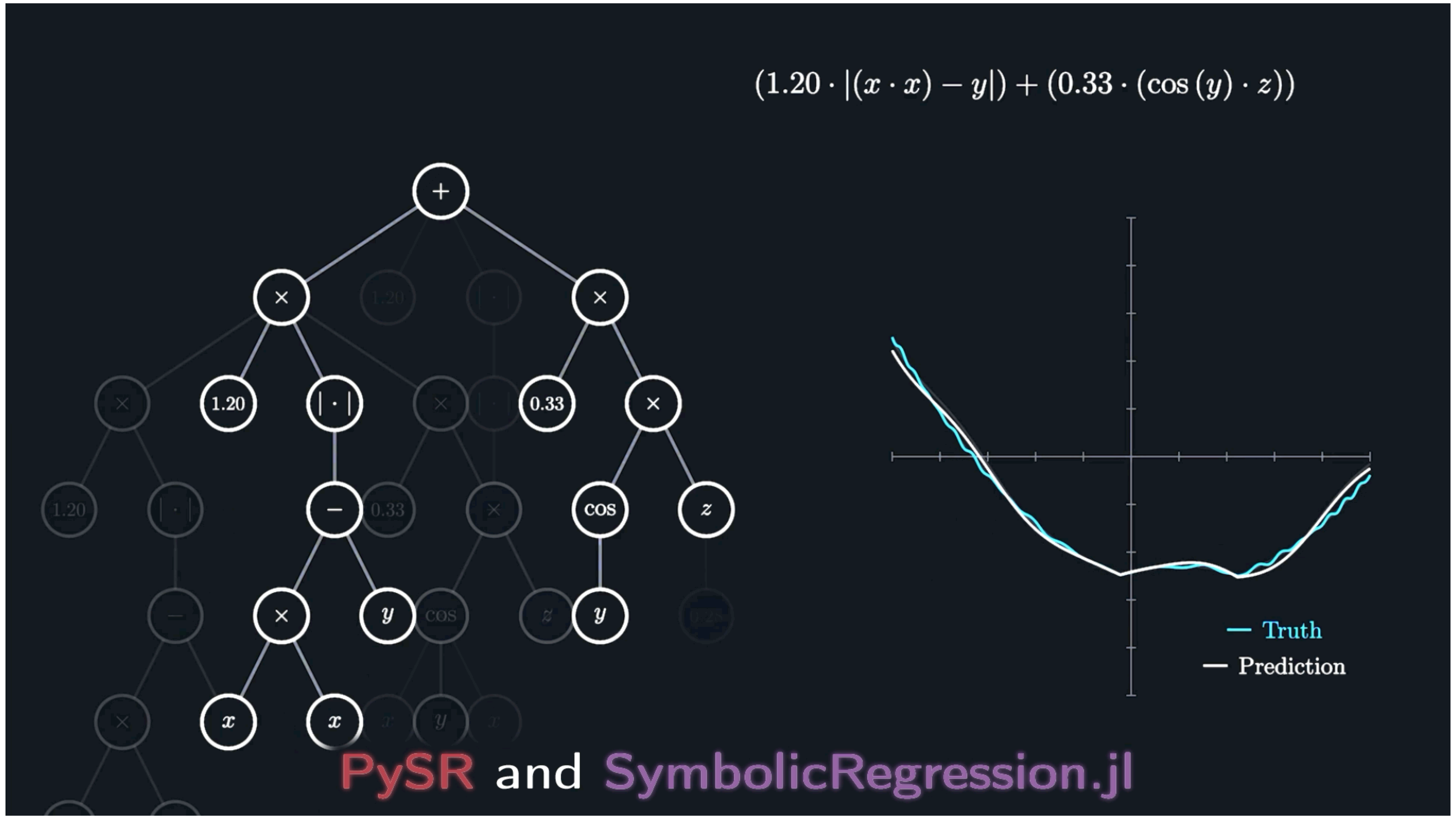
Using Symbolic Regression to Analyse Plasma Data

How PySR works:

PySR is a Symbolic Regression that uses any unary or binary operators to return the best fit it finds with those operators.

PySR is an evolutionary algorithm trained with “natural” selection. The different populations can also interact with exchanges of branches.

The algorithm, then, chooses the equation based on a the log-loss complexity curve, meaning a lower complexity may be chosen even with a higher loss.



$(1.20 \cdot |(x \cdot x) - y|) + (0.33 \cdot (\cos(y) \cdot z))$
+
×
×
1.20
|·|
0.33
×
−
cos
z
×
y
y
x
x
Truth
Prediction
PySR and SymbolicRegression.jl

How PySR actually works

To understand how to work with the library, I first did a scan with the main parameters PySR uses:

- **Seed**
  For a practical number of iterations (<100) there is a high dependency on the chosen seed.

- **Error**
  PySR has no problem dealing with small errors ($\lesssim 1\%$). With denoise = True we can deal with higher errors ($\lesssim 10\%$)

- **Iterations and populations**
  A good number os populations is 2x or 3x the amount of cores available. Of course the amount of Iterations will improve the fit, usually until a certain point. What I like to do is set the iterations to 10.000.000 and set a maximum amount of time (e.g. 2 hours)

- **Number of points**
  A good amount of data is 1000 points. For less data the algorithm can also perform well but not as good. If the data is much larger than 1000 it can be a good idea to divide the dataset in sets of data with 1000 points (to run in parallel and to have more populations)

Artificial Data

Real Data

Fitted equations:

$$\frac{e^{-\frac{(x-x_0)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$$

$$E\frac{\partial F}{\partial u} - \frac{\partial F}{\partial x}v$$

Next steps

Next I intend to analyse different waves in plasma with data created from ZPIC.
This gives us a hierarchy of expressions to analyse with PySR and check PySR's utility in analysing real data.
For this I intend to try to find the equations for 5 different seeds.

	ZPIC equation	Correct Solutions
Plasma Oscillations	$\omega^2 = 1 + 0.0027\,k^2$	$^{n}/_{5}$
Ordinary Waves	$\omega = \sqrt{1 + k^2}$	$^{n}/_{5}$
Right Waves	$k^2 = \omega^2 - \dfrac{\omega}{\omega - 2}$	$^{n}/_{5}$
Left Waves	$k^2 = \omega^2 - \dfrac{\omega}{\omega + 2}$	$^{n}/_{5}$
Extraordinary waves	$\omega_{\pm} = \dfrac{k^2 + 3 \pm \sqrt{k^4 - 2k^2 + 5}}{2}$	$^{n}/_{5}$