https://doi.org/10.15302/frontphys.2025.014209

RESEARCH ARTICLE

Boosting SISSO performance on small sample datasets by using random forests prescreening for complex feature selection

Xiaolin Jiang, Guanqi Liu, Jiaying Xie, Zhenpeng Hu†

School of Physics, Nankai University, Tianjin, 300071, China

Corresponding author. E-mail: †zphu@nankai.edu.cn

Received May 29, 2024; accepted August 30, 2024

Supporting Information

1 Random Forests algorithm for feature screening

Gives a set of material datasets $D = \{f_1, P_1\}, \ldots, \{f_N, P_N\}$, where $f_1, \ldots, f_N \in \mathbb{R}^M, P_1 \ldots P_N \in \mathbb{R}$, that is N materials are provided, each of which has M features. The first screening of features is undertaken using random forest. Random Forests [1] is a machine learning algorithm that integrates multiple trees with the idea of integrated learning, where the basic unit is the decision tree. Simply compare how much each feature contributes to each decision tree in the random forest, the larger the contribution, the greater the impact it has on the outcome. The contribution size is measured by the Gini index and the out-of-bag data error rate. The Gini value of the dataset D can be defined as

$$Gini(D) = \sum_{k=1}^{k} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{k} p_{k^2} \quad , \tag{1}$$

where assume that the value space of P is $\{s_1, \ldots, s_k\}$, pk denotes the probability that P takes s_k. The larger the $Gini(D)$, the greater the uncertainty of dataset D.

The Gini index corresponding to characteristic F_m be defined as

$$Gini\_index(D, F_m) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v) \, , \tag{2}$$

where feature F_m has V possible values $\{a_1, \ldots, a_v\}$,, and when the dataset D is partitioned using feature F_m, V subsets are generated. The $v\text{-}th$ subset containing all the samples in D with a_v values on the feature F_m in D, and is marked as D^v. $|D^v|$ denotes the number of samples when feature $F_m = a_v$. The smaller the $Gini\_index(D, F_m)$, the more the data uncertainty in the dataset is reduced by dividing by F_m, the more important F_m is for the result. Assuming that the random forest consists of t trees, then the importance score of Fm can be calculated using Eq. (3):

$$I(F_m) = \frac{\sum_{i=1}^{t} Gini\_index(D_i, F_i)}{\sum_{m=1}^{M} \sum_{i=1}^{t} Gini\_index(D_i, F_i)} \quad . \tag{3}$$

However, the data set is very small. Due to the sample division and the selection of the corresponding features of the random forest are all random, the prediction accuracy rate obtained from each time cross-validation will be different, and the difference may be very large. Only one Random Forests calculation is conducted, and there is experimental contingency. Therefore, we need to repeat R (R > 50 generally) times of random forest feature evaluation, and each time there will be a score $I_r(F_m)$ of feature F_m and a prediction accuracy G_r obtained from cross-validation. The combined importance score for feature F_m can be expressed as

$$I(F_m) = \frac{\sum_{r=1}^{R}[I_r(F_m), G_r]}{\sum_{m=1}^{M}\sum_{r=1}^{R}[I_r(F_m), G_r]} \quad . \tag{4}$$

After obtaining the scores for each feature, rank them from largest to smallest. Because the scores have been normalized, the importance of each feature can be clearly compared. This paper plans to select the top half of the features as important features for SISSO calculation.

2 SISSO algorithm for symbolic regression

Once the important features have been initially screened, SISSO can be used to filter the key features. To give a non-linear relationship between the material property "P" and the feature "f", the feature vector f is mapped in a higher dimensional space by a function $H: \mathbb{R}^{M_1} \to \mathbb{R}^{M_2}$. This is done by using the starting points ϕ_0', which indicate the important features screened from random forest, and then recursively performing algebraic combination of algebraic operations on these starting features to expand the features space. The operators set can be defined as

$$H^{(m)} \equiv \{I, +, -, \times, \div, exp, log, |-|, \sqrt{\ }, \ ^{-1}, -\ ^{2}, |\ ^{3}\}[\varphi_1, \varphi_2] \ , \tag{5}$$

where φ_1 and φ_2 are components in the features space, and the superscript (m) indicates that a dimensional analysis is performed to preserve only meaningful combinations. In each iteration, $H^{(m)}$ will operate on all available combinations, and the features space will grow recursively, resulting in the features space ϕ:

$$\phi \equiv \bigcup_{q=1}^{Q} \widehat{H}^{(m)}[\varphi_1, \varphi_2], \forall \varphi_1, \varphi_2 \in \phi_{q-1} \ \ , \tag{6}$$

The quantity of elements in ϕ grows quickly with the number of iterations, and the number of candidate features M also increases rapidly. In order to find the key features affecting P, using the principle of CS, an attempt is made to find $c \in \mathbb{R}^M$ with the least non-zero terms satisfying $Fc = P$, which can be written as

$$\underset{\boldsymbol{c}:\boldsymbol{Fc}=\boldsymbol{P}}{\operatorname{argmin}} \|\boldsymbol{c}\|_0 \ , \tag{7}$$

where $P = (P_1, \ldots, P_N)^T$ is the output property column vector $F = \{f_1, \ldots, f_N\}^T$ is the input $(N \times M)$ dimensional feature matrix, $\|c\|_0$ is the number of non-zero elements of vector c. c is n-sparse, it means that $\|\boldsymbol{c}\|_0 \leq n$. However, there is a significant disadvantage: when M is very large, that is when the number of reference features is very large, it is computationally infeasible because solving Eq. (7) is an NP-hard problem and only exhaustive methods can be used to find the solution c. Meanwhile, CS requires [3,4]: to restore a stable unique n-sparse c from F and P, it must also satisfy:

$$N \geq Cn \ln M \tag{8}$$

where C is a constant, usually between 1 and 10 [5]. This limits the number of features M to e^{N/C_n} or less. But the huge features space obviously no longer meets the requirements of the effectiveness of CS.

To solve the problem of the large features space, SISSO introduced sure independence screening (SIS). It makes

$$\omega = PF^T \ , \tag{9}$$

where $\omega = (\omega_1, \ldots, \omega_M)^T$ and $\boldsymbol{F} = \{\boldsymbol{F_1}, \ldots, \boldsymbol{F_M}\}$. If ω_i is larger, then F_i is more strongly correlated with P. $|\omega_i|$ can be sorted from largest to smallest so that a features subspace S_{1D} of an appropriate size can be selected, where F_{1D} is the column of F with the highest correlation with P. If you only want to find a relationship between the property and a combination of features, this is actually already obtained.

If one wishes to obtain $n(n > 1)$ dimensional descriptors, which means a relationship between the property defined in terms of n sets of feature combinations. There also needs to define the residuals of the n-dimensional descriptors as

$$\Delta_{nD} \equiv P - F_{nD} c_{nD} \, , \tag{10}$$

where F_{nD} is the column whose correlation with P is greatest in the $n\text{-}th$ SIS, $C_{nD} = {F_{nD}}^T {F_{nD}}^{-1} {F_{nD}}^T P$ is the least-squares solution of the F_{nD} to P fit. Then SIS selects the features subspace S_{nD} which has the largest correlation with $\Delta_{(n-1)D}$ and combines S_{nD} with $S_{(n-1)D}$ to form S_n. The size of the S_n feature space is required to satisfy the requirement that a stable sparse solution c can be solved.

Finally, CS is exploited in the feature space S_n to obtain the best n-dimensional descriptor between a few key features and the property. The flow diagram of the overall algorithm is shown in Fig. S1.

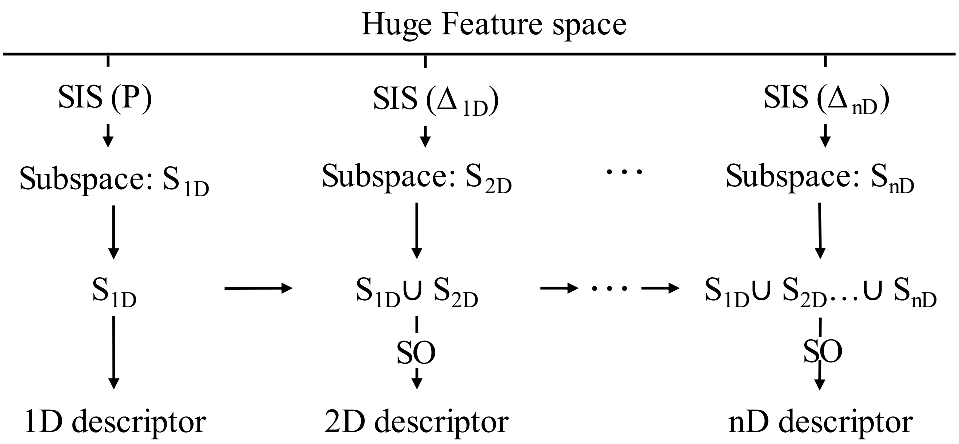

Fig. S1 Flowchart of SISSO algorithm.

3 Result

Table S1 Classification performance of random forests algorithm.

Random Forests (8 features)			
Number of samples	Training_Accuracy	Testing_Accuracy	Training_Time
45	1.000	0.910	< 1 S
75	1.000	0.911	< 1 S
150	0.980	0.933	< 1 S
224	0.978	0.893	< 1 S

Table S2 Two-dimensional descriptors derived by symbolic regression using SISSO and RF-SISSO in 224-sample subset.

Datasets		SISSO	RF-SISSO
1st	d1	$d_{AB} \times (V + nB) \times \lvert IE_B - \chi_A/nA \rvert$	$[(\nu_A + \nu_B)/V] \times \exp(\chi_A - \chi_B)$
	d2	$(rcov_A/CN_A)/(nA/nB - \chi_A/\chi_B)$	$\exp(V/\chi_B)(IE_B + \lvert EA_B - \chi_A \rvert)$
2nd	d1	$[(V/IE_B)/\chi_A]/(V + rcov_A/rcov_B)$	$(V/\chi_A)^3 \times \nu_A/(\chi_B)^3$
	d2	$\lvert d_{AB}/rcov_A \rvert - (V/nA) \times (nB)^2$	$(EA_A - 2IE_B - \chi_A)/(\chi_A/V)$
3rd	d1	$(\chi_A + \chi_B)/(\chi_A/V + nB \times \chi_A)$	$V \times \chi_B \times [\exp(-\nu_A) + \exp(-\chi_A)]$
	d2	$\lvert \chi_A/V \rvert - (IE_A - \chi_B) \times \lvert nA - nB \rvert$	$\mathrm{V} \times (EA_A + IE_B) \times (\lvert \nu_A - \nu_B \rvert - \nu_A)$
4th	d1	$d_{AB} \times (V + nB) \times (\chi_A/nA - IE_B)$	$V \times \chi_B \times [\exp(-\nu_A) + \exp(-\chi_A)]$
	d2	$\lvert \exp(-\chi_A \nu_B) - \exp(-\lvert CN_A - CN_B \rvert) \rvert$	$(IE_B - 2\chi_B)/(\chi_A/V)$

Table S3 Two-dimensional descriptors derived by symbolic regression using SISSO and RF-SISSO in 150-sample subset.

Datasets		SISSO	RF-SISSO
1st	d1	$(V - nA)[IE_B/\chi_A - d_{AB}/rcov_B]$	$[(\nu_A + \nu_B)/\chi_B] \times (\chi_A/V)^2$
	d2	$(CN_A \times d_{AB})^3 \times \exp(\chi_B/nB)$	$(EA_A - IE_B - \lvert IE_B - \chi_B \rvert)/(\chi_A/V)$
2nd	d1	$V/\exp(CN_B) + \chi_A/(V \times \chi_B)$	$(IE_B \times \chi_A)/((IE_B - \chi_B) \times V \times \chi_B)$
	d2	$(CN_A \times IE_B)^3 + \lvert CN_A - V \times CN_B \rvert$	$V^9 \times \nu_A \times \exp(\nu_B)$
3rd	d1	$(\chi_A + \chi_B)/(\chi_A/V + nB \times \chi_A)$	$V \times IE_B \times [\exp(-\nu_A) + \exp(-\chi_A)]$
	d2	$CN_A \times IE_A \times \lvert nB \times CN_B - \lvert CN_A - CN_B \rvert \rvert$	$\lvert \lvert EA_B - \chi_A \rvert - EA_B \rvert - \lvert 2\chi_A - \chi_B \rvert$
4th	d1	$\lvert \chi_B - IE_A/V \rvert - \lvert EA_B - \chi_B/V \rvert$	$(2EA_A + IE_B - EA_B)/(\chi_A/V)$
	d2	$(CN_B \times IE_B)^3 \times (EA_B \times \chi_A)/nA$	$EA_B + \chi_A + (\chi_B)^3/(IE_B \times \chi_A)$

Table S4 Two-dimensional descriptors derived by symbolic regression using SISSO and RF-SISSO in 75-samples subset.

Datasets		SISSO	RF-SISSO
1st	d1	$(IE_B - \chi_A)/(nB \times \lvert \chi_B - IE_A/V \rvert)$	$[(V \times \chi_B)^3 \times \nu_B]/(V \times \chi_A)^3$
	d2	$(V \times rcov_A)/nB \times (IE_B + \chi_B/nA)$	$[(IE_B)^3 \times (\nu_A)^4]/(EA_A - EA_B)$
2nd	d1	$(V \times \chi_B - IE_A) \times \exp(CN_A/nB)$	$[(IE_B \times \chi_B)^3 \times (V)^3]/(\chi_A)^3$
	d2	$(V \times \chi_B - IE_A) \times \exp(CN_A + CN_B)$	$(V/\chi_A)^3 \times (\chi_B)^6$
3rd	d1	$(V)^2 \times \chi_B \times [(IE_B \times d_{AB})^3]$	$(V \times IE_B)^3 \times (EA_B + \chi_B)/\chi_A$
	d2	$(V)^2 \times IE_B \times [(IE_B \times d_{AB})^3]$	$(V/\chi_A)(IE_B \times \chi_B)^3$
4th	d1	$(\chi_A/nA - IE_B) \times (V \times d_{AB} + rcov_A)$	$V \times IE_B \times [\exp(-\nu_A) + \exp(-\chi_A)]$
	d2	$V \times nA \times \chi_B \times (d_{AB} \times IE_B)^2$	$[V \times (IE_B + \chi_B)]/\exp(V/\nu_B)$

Table S5 Two-dimensional descriptors derived by symbolic regression using SISSO and RF-SISSO in 45-sample subset.

Datasets		SISSO	RF-SISSO
1st	d1	$(V \times \chi_B)^3 \times \exp\,(IE_A/\chi_A)$	$[(V \times \chi_B)^3 \times IE_B]/\nu_A$
	d2	$\exp\,(IE_A/\chi_A + V \times \chi_B)$	$(IE_B \times \chi_B \times V^2)/{\chi_A}^2$
2nd	d1	$(V \times \chi_B - IE_A) \times \exp\,(CN_A/nB)$	$(V/\chi_A)^3 \times {\chi_B}^6$
	d2	$[\chi_A/(V \times nB)] \times \exp\,(CN_B/nA)$	$(V/\chi_A)^3 \times (EA_A \times \chi_B)^6$
3rd	d1	$[\chi_A/(V \times \chi_B)] \times \exp\,(CN_B/nA)$	$(V)^3 \times IE_B \times (\chi_A - \chi_B)$
	d2	$(V \times IE_B)^3 \times CN_B \times (\chi_A - \chi_B)$	$IE_B \times \chi_B \times (V/\chi_A)^3$
4th	d1	$\chi_B \times (V - nB) \times (d_{AB} \times IE_B)^3$	$V/(EA_B/IE_B + \chi_A/\chi_B)$
	d2	$(d_{AB} \times IE_B)^3 \times (V \times \chi_B)/CN_A$	$(\nu_A)^3/(EA_A + EA_B - \chi_A)$

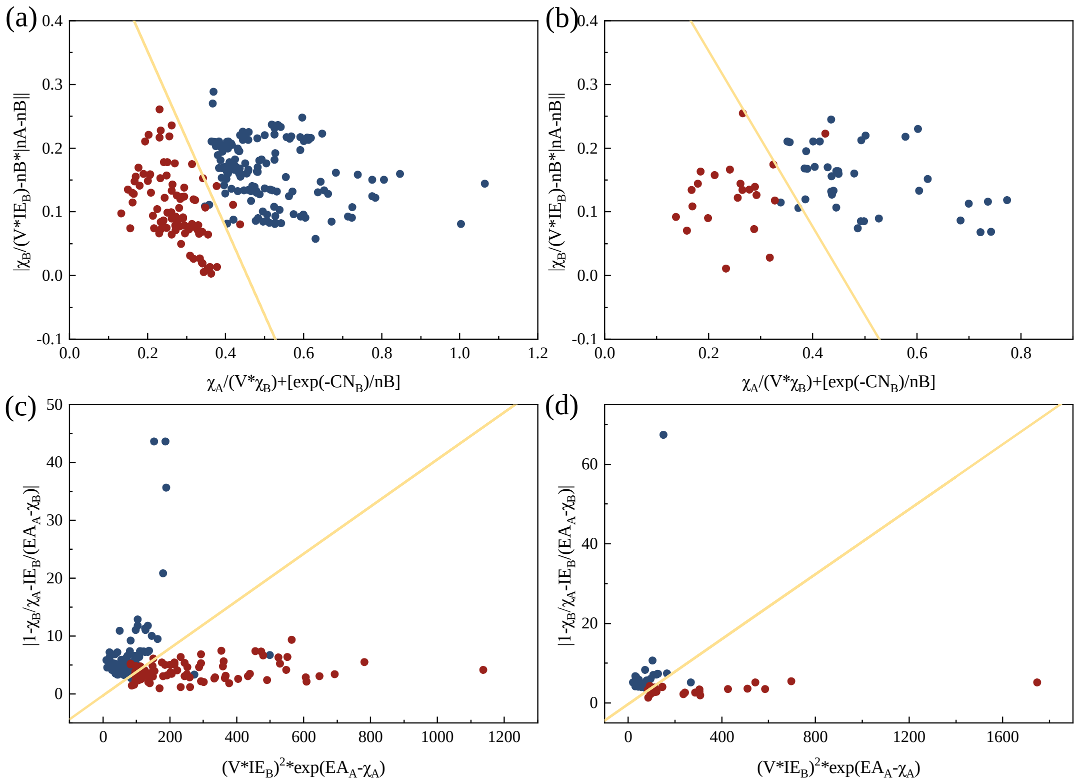

Fig. S2 Visualizations of SVC classification for 2D descriptors based on SISSO training in a 224-sample subset. **(a)** training and **(b)** testing sets with operators from SISSO; **(c)** training and **(d)** testing sets with operators from RF-SISSO.

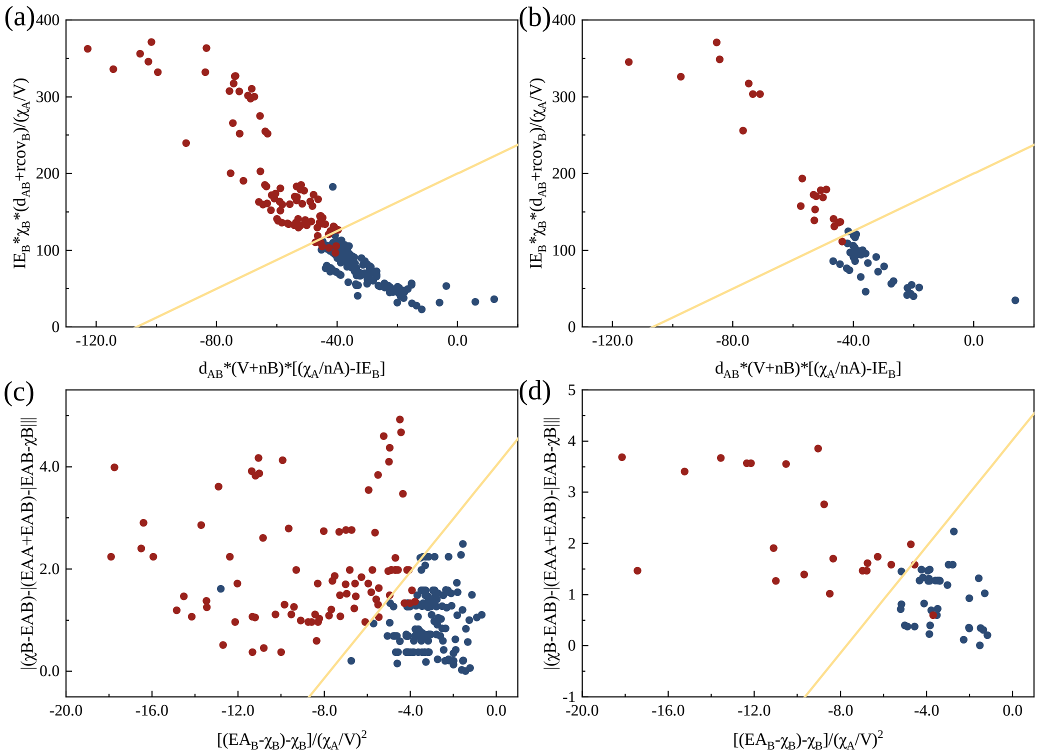

Fig. S3 SVC classification visualizations for 2D descriptors based on SISSO training in a 150-sample subset. **(a)** training and **(b)** testing sets with operators from SISSO; **(c)** training and **(d)** testing sets with operators from RF-SISSO.

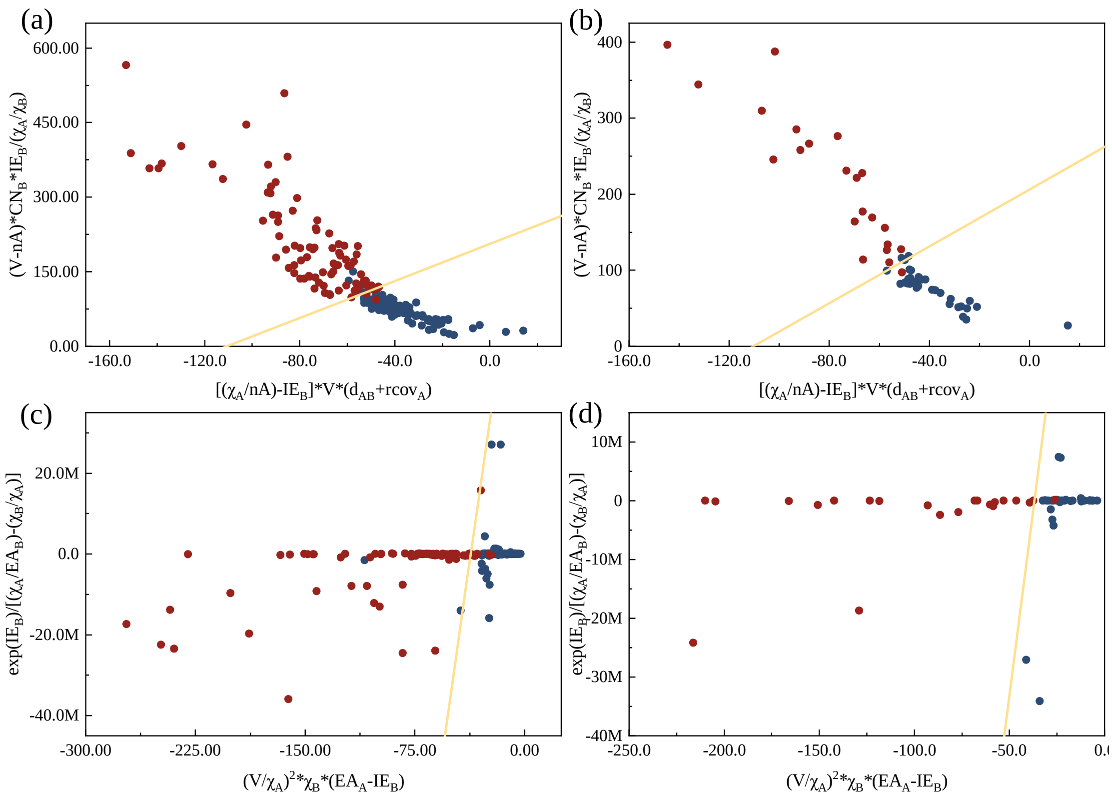

Fig. S4 SVC classification visualizations for 2D descriptors based on SISSO training in a 75-sample subset. **(a)** training and **(b)** testing sets with operators from SISSO; **(c)** training and **(d)** testing sets with operators from RF-SISSO.

Table S6 Comparison of regression efficiency and accuracy between RF-SISSO and SISSO on small sample sizes.

System	Algorithm	r_train	RMSE_train	Time (s)	r_test	RMSE_test
PFC-20	RF-SISSO	0.999	0.001	5417.28	0.817	0.022
	SISSO	0.999	0.001	15454.56	0.847	0.012
PFC-40	RF-SISSO	0.999	0.002	599.6	0.972	0.006
	SISSO	0.999	0.002	11062.56	0.849	0.013
PO3-20	RF-SISSO	0.998	0.007	2686.08	0.962	0.080
	SISSO	0.997	0.007	10811.44	0.883	0.079
PO3-40	RF-SISSO	0.998	0.010	6616.8	0.948	0.066
	SISSO	0.998	0.010	15885.6	0.885	0.110

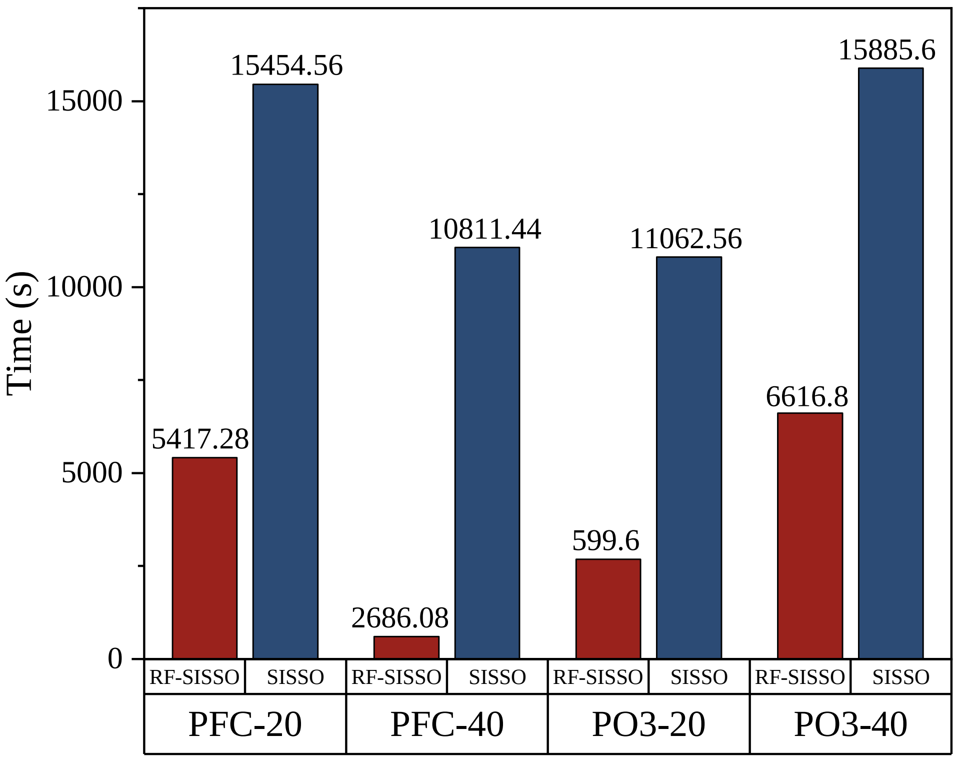

Fig. S5 Regression time of SISSO and RFSISSO for different sample numbers in PFC and PO3 system.