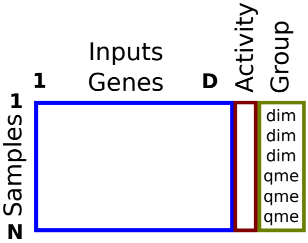
Cross-validation for comparing qSIP prediction models trained on same or other groups

Toby Dylan Hocking
toby.hocking@nau.edu
toby.hocking@r-project.org

January 11, 2024

Machine learning predictive analysis of qSIP data

- Inputs/features $\mathbf{x} \in \mathbb{R}^D$ is vector of TODO for D genes (Amplicon Sequence Variants / ASVs, range from 0 to 10).
- Output $y \in \mathbb{R}$ is relative activity/growth per day from qSIP (excess atom fraction/EAF normalized by maximum isotope enrichment and incubation length, ranging from 0 to 0.3315).
- Want to learn $f(\mathbf{x}) = y$ (predict growth from genes).
- Hypothesis: expect we can learn f on mixed conifer (MC) controls in experiment=dim (room temp), and accurately predict experiment=qme at temp=15C (or vice versa). TODO Jeff what is qme/dim?
- Question: is this expectation consistent with the data?
- Answer by using 10-fold cross-validation: train on one experiment or other, quantify prediction error on held out test set.



Inputs

Genes

1

D

Activity

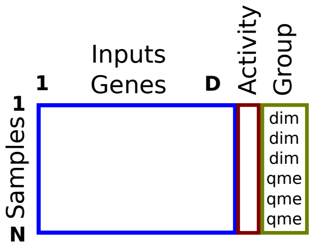
Group

1

N

Samples

dim
dim
dim
qme
qme
qme



Inputs

Genes

1

D

Activity

Group

1

Samples

N

dim

dim

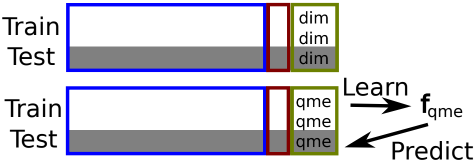
dim

qme

qme

qme

Same



Train

Test

dim

dim

dim

Train

Test

qme

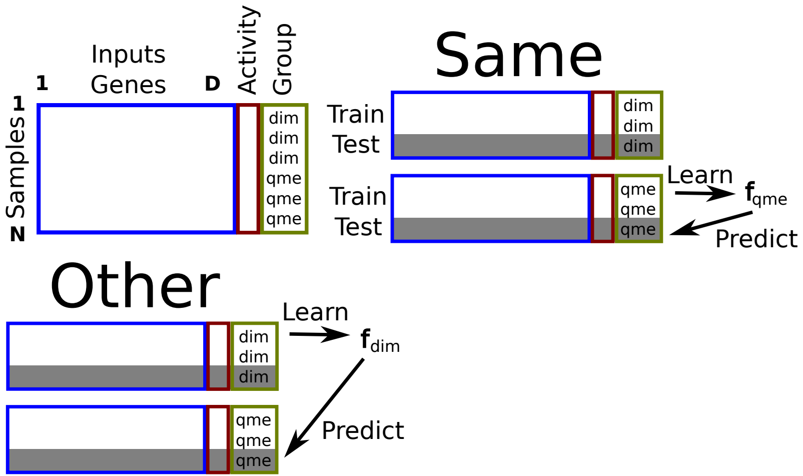
qme

qme

Learn

$\mathbf{f}_{qme}$

Predict



Inputs

Genes

1

D

Activity

Group

Samples

1

N

dim
dim
dim
qme
qme
qme

Same

Train

Test

dim
dim
dim

Train

Test

qme
qme
qme

Learn $\mathbf{f}_{\mathrm{qme}}$

Predict

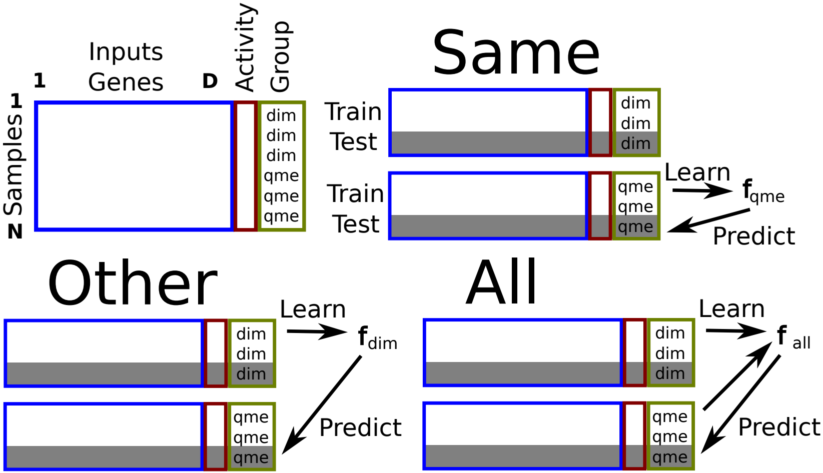
Other

dim
dim
dim

Learn $\mathbf{f}_{\mathrm{dim}}$

qme
qme
qme

Predict



Inputs

Genes

1

D

Activity

Group

1

Samples

N

dim
dim
dim
qme
qme
qme

Same

Train

Test

dim
dim
dim

Train

Test

qme
qme
qme

Learn

f_{qme}

Predict

Other

dim
dim
dim

Learn

f_{dim}

qme
qme
qme

Predict

All

dim
dim
dim

Learn

f_{all}

qme
qme
qme

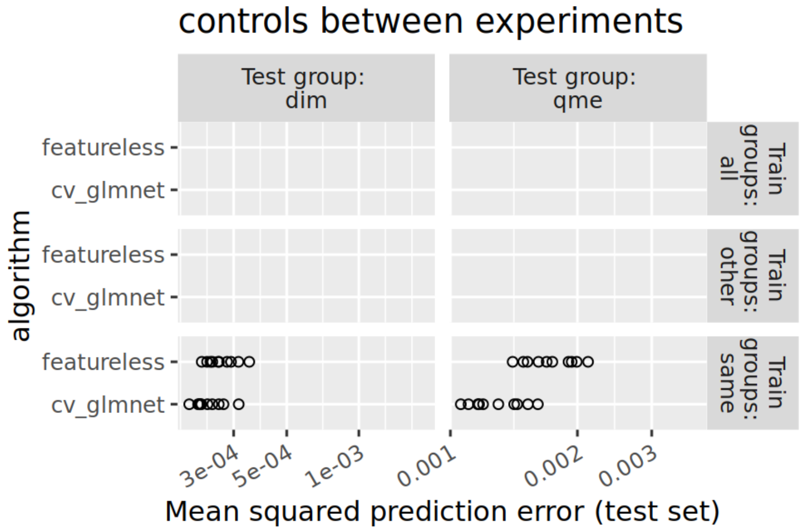
Predict

Comparison 1: controls in different experiments

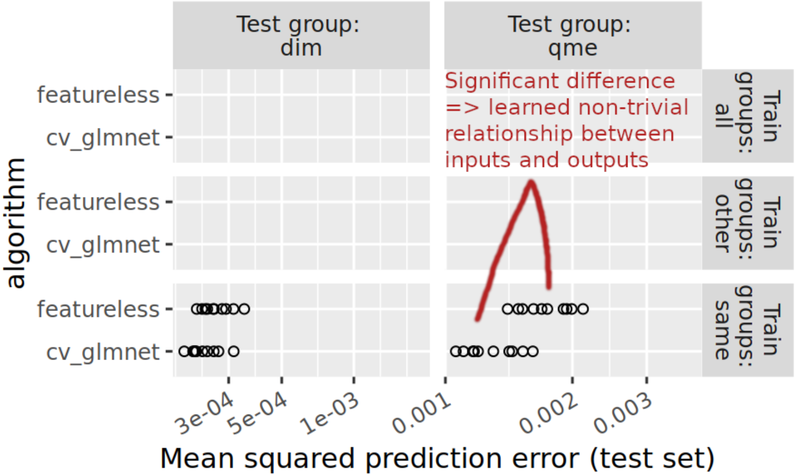
- Data table with $N = 7710$ rows/observations (TODO), across two experiments dim=3120, qme=4590.
- $D = 8380$ gene features.
- We compare two learning algorithms

  cv_glmnet: L1 regularized linear model (LASSO), small subset of important genes selected and used for prediction (other un-important genes are not used for prediction).

  featureless ignore all genes/features, and always predict mean output in train set.
- If there is any non-trivial relationship/pattern learned between inputs and outputs, then **linear model should have smaller prediction error than featureless**.
- If patterns are similar in different groups/experiments (dim and qme), then **linear model should have similar prediction error, when trained on other groups/experiments**.

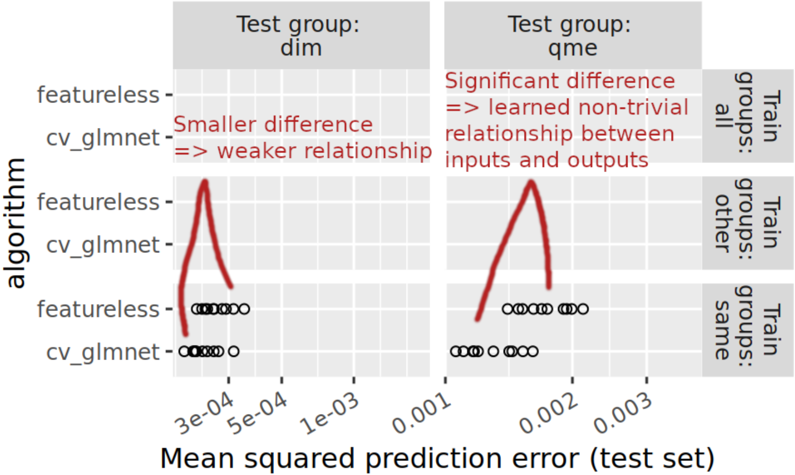

controls between experiments
Test group:
dim
Test group:
qme
Train
groups:
all
Train
groups:
other
Train
groups:
same
algorithm
featureless
cv_glmnet
featureless
cv_glmnet
featureless
cv_glmnet
3e-04
5e-04
1e-03
0.001
0.002
0.003
Mean squared prediction error (test set)

controls between experiments



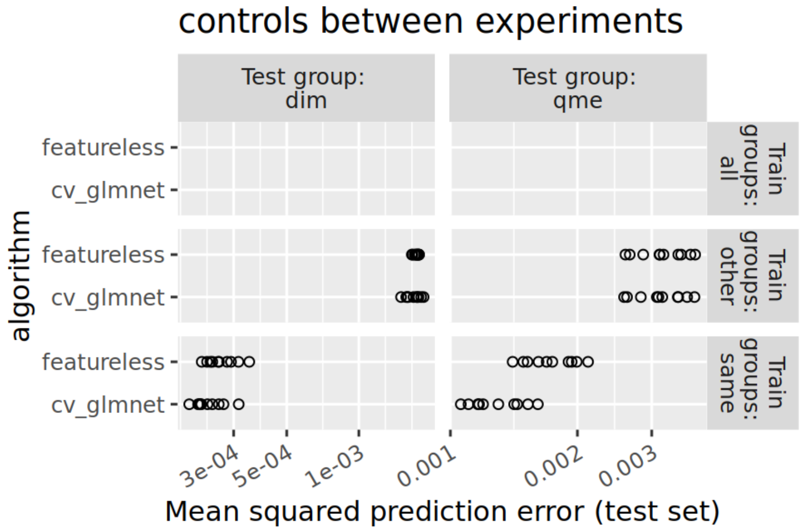
Significant difference => learned non-trivial relationship between inputs and outputs

controls between experiments

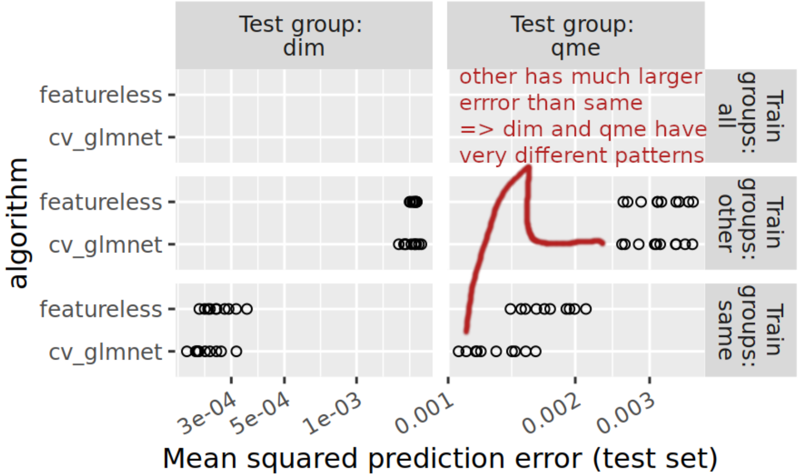


Smaller difference
=> weaker relationship

Significant difference
=> learned non-trivial
relationship between
inputs and outputs

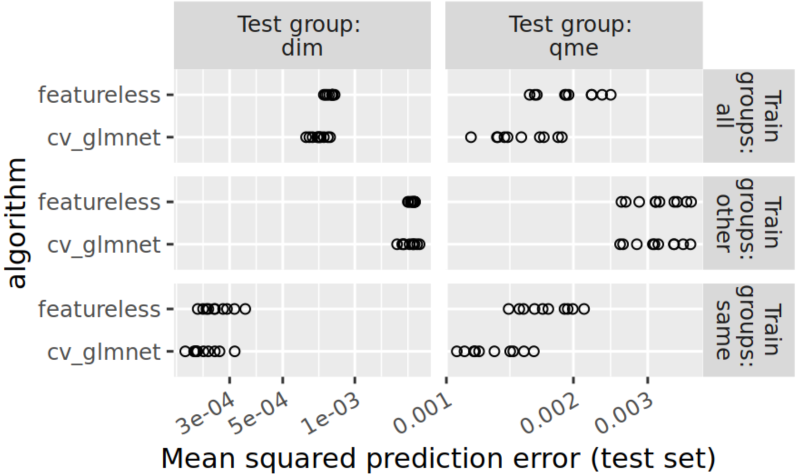

controls between experiments
Test group:
dim
Test group:
qme
Train
groups:
all
Train
groups:
other
Train
groups:
same
featureless
cv_glmnet
featureless
cv_glmnet
featureless
cv_glmnet
algorithm
3e-04
5e-04
1e-03
0.001
0.002
0.003
Mean squared prediction error (test set)

controls between experiments



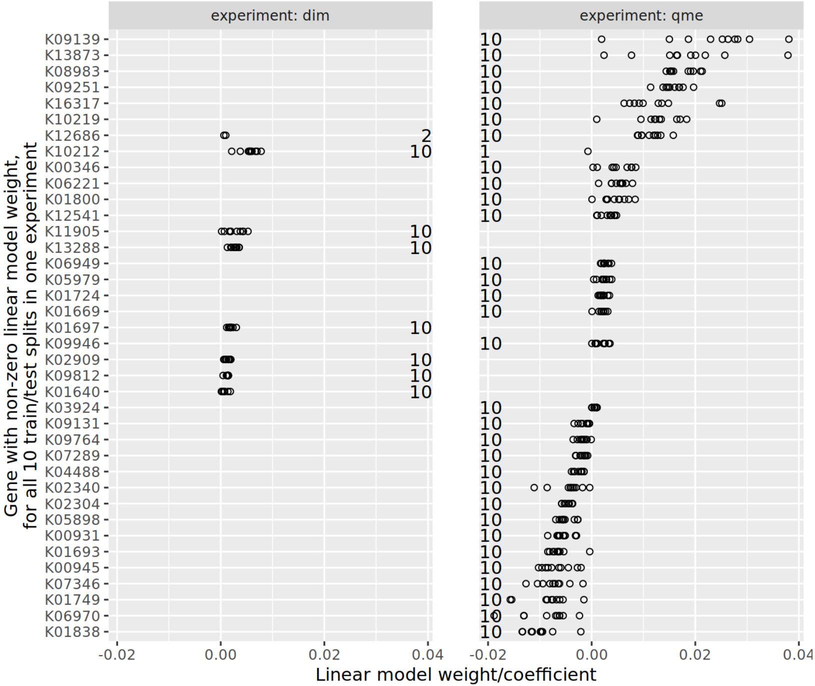
other has much larger errror than same => dim and qme have very different patterns

controls between experiments



Interpretation of linear model prediction error and weights/coefficientsc

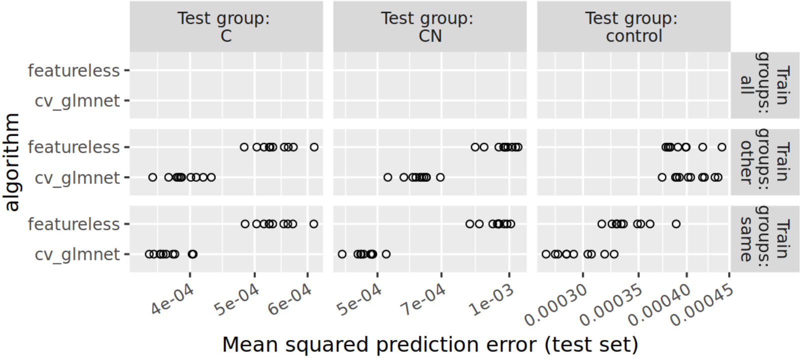
- Hypothesis was: expect we can learn f on mixed conifer (MC) controls in experiment=dim (room temp), and accurately predict experiment=qme at temp=15C (or vice versa).
- Prediction error cross-validation analysis is not consistent with that hypothesis.
- So there should be a different prediction function in each experiment, what is the difference?
- The L1 regularized linear model (LASSO) can be interpreted in terms of which genes are important/used for prediction (non-zero weights/coefficients) and others are ignored (weights=0, not used for prediction).
- Compute and plot weights which are non-zero/important in all 10 train/test splits of cross-validation.

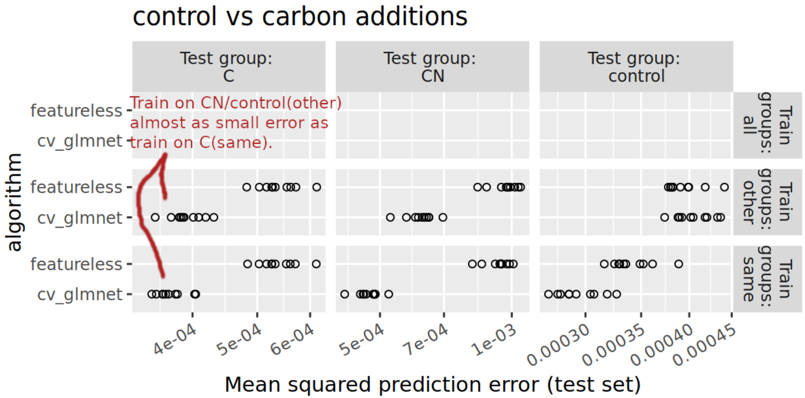

experiment: dim
experiment: qme
Gene with non-zero linear model weight,
for all 10 train/test splits in one experiment
K09139
K13873
K08983
K09251
K16317
K10219
K12686
K10212
K00346
K06221
K01800
K12541
K11905
K13288
K06949
K05979
K01724
K01669
K01697
K09946
K02909
K09812
K01640
K03924
K09131
K09764
K07289
K04488
K02340
K02304
K05898
K00931
K01693
K00945
K07346
K01749
K06970
K01838
2
10
10
10
10
10
10
10
10
10
10
10
10
10
10
1
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
-0.02
0.00
0.02
0.04
-0.02
0.00
0.02
0.04
Linear model weight/coefficient

Comparison 2: control versus carbon additions

- $N = 60877$ samples total, in 3 groups/treatments: control=17225, C=23214, CN=20438.
- Same $D = 8380$ gene features.
- Can we train on one group/treatment, and predict accurately on another?

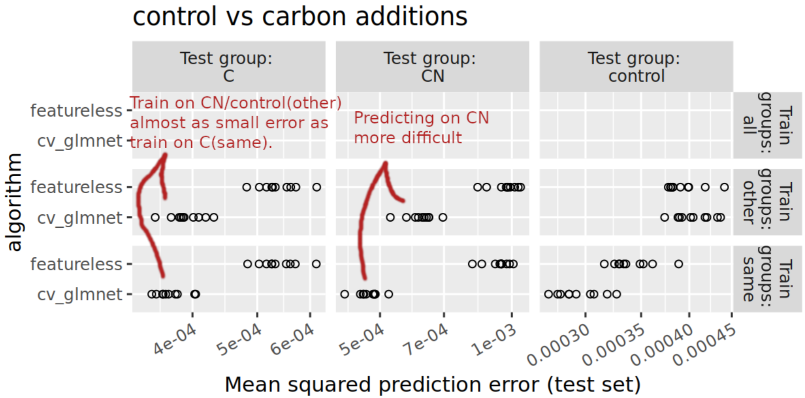
control vs carbon additions





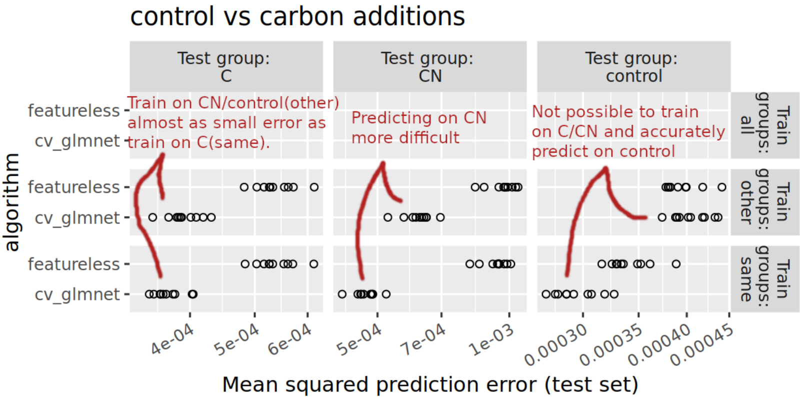
control vs carbon additions

Train on CN/control(other) almost as small error as train on C(same).



Train on CN/control(other) almost as small error as train on C(same).

Predicting on CN more difficult



control vs carbon additions

Train on CN/control(other) almost as small error as train on C(same).

Predicting on CN more difficult

Not possible to train on C/CN and accurately predict on control

Discussion and conclusions

- TODO
- Free/open-source software available: mlr3resampling R package.