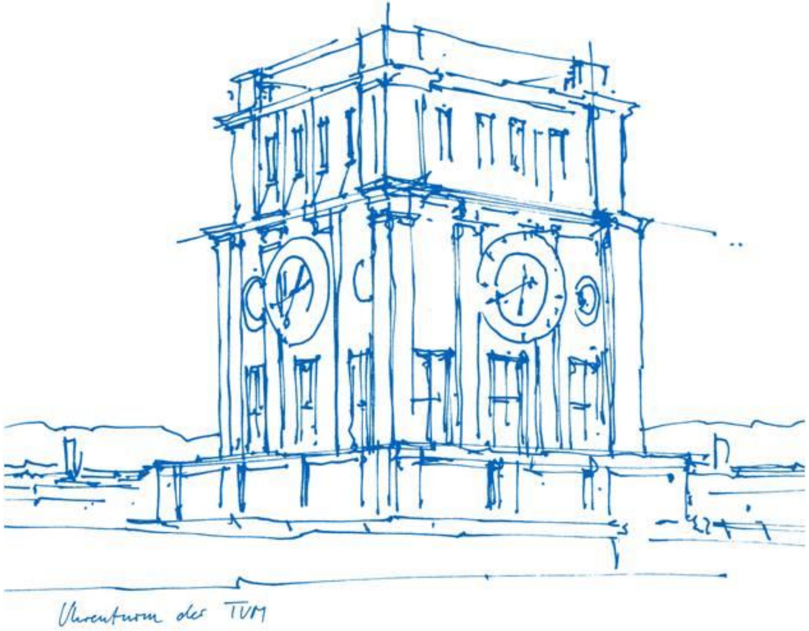
CRISPR Toolbox - a deep learning approach to improve CRISPR/Cas experiments

TUM Data Innovation Lab

25. February 2022



Team: Daria Yasafova, Dennis Gankin, Yevhenii Sharapov,
Chelsea Bright, Firas Driss, Francesco Campi

Mentors: Dr. Lisa Barros de Andrade e Sousa, Dr. Erinc Merdivan
Project Lead: Dr. Ricardo Acevedo
Supervision: Prof. Massimo Fornasier

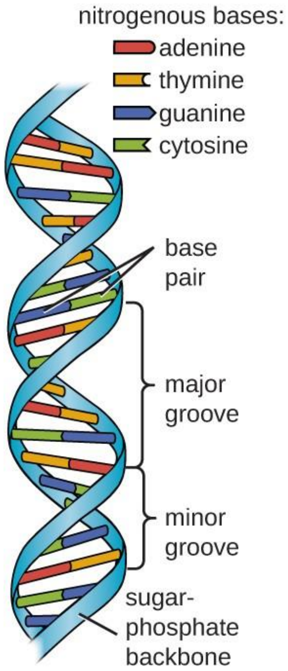
Agenda

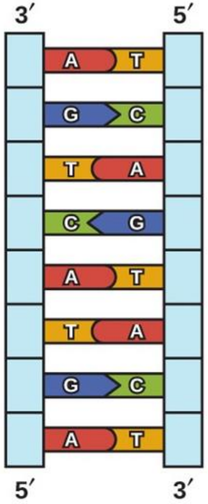
Biological background

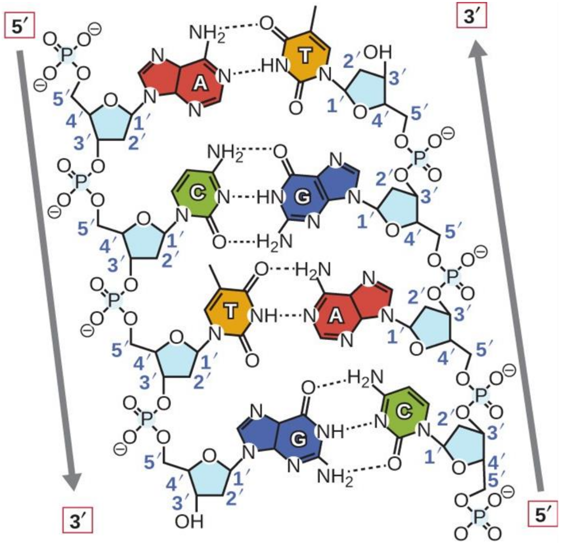
panCRISPR tool and its modules

Discussion and outlook

DNA (Deoxyribonucleic Acid)

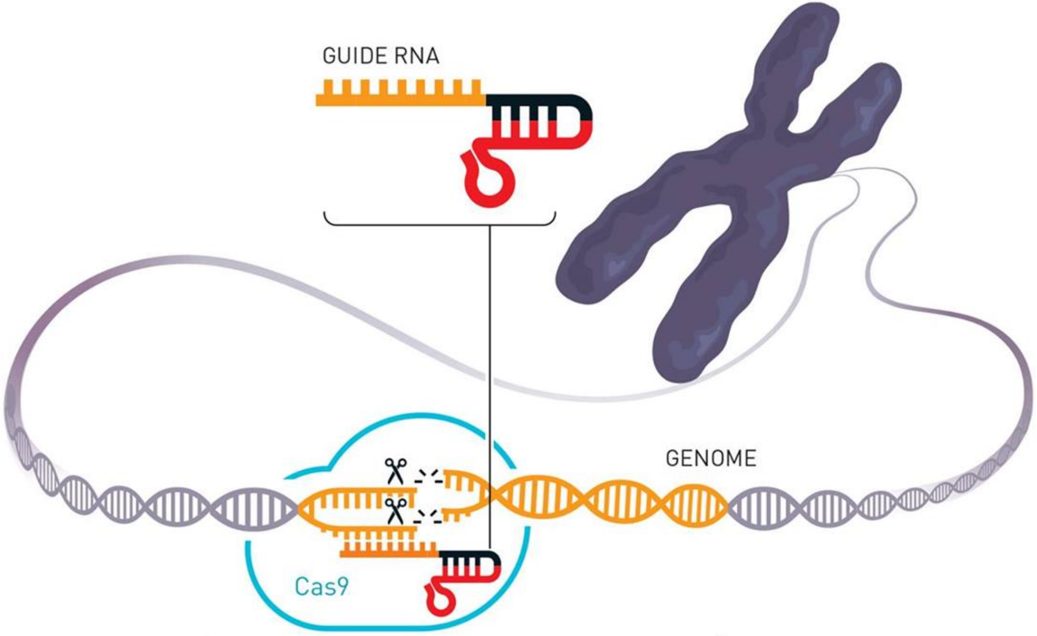






pictures: https://steemit.com/dna/@patelchirag/dna-structure

CRISPR/Cas



picture: Royal Swedish Academy of Sciences

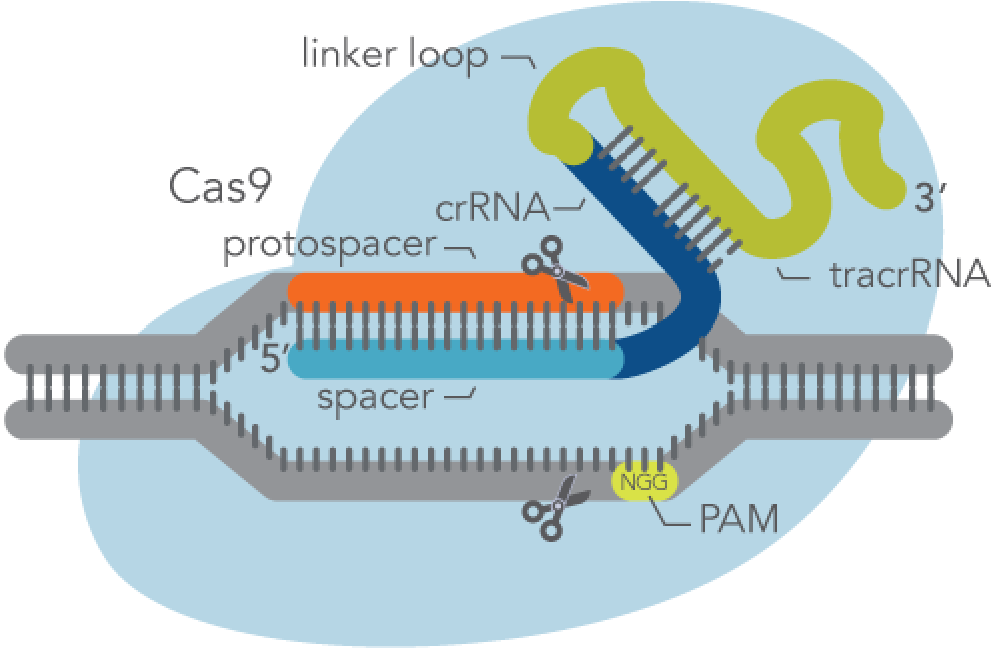
Guide design

efficiency

binding to desired site
with high probability (on-target)

specificity

unlikely to bind to other sites
in the genome (off-target)



State-of-the-art guide design tools

On-target tools

- CRISPRon
- CRISPRater
- CRISPRpred
- DeepCpf1

methods: rule based, SVM, deep models etc.

Off-target tools

- CRISPRoff
- Cas-OFFinder
- MIT
- FlashCry

methods: search based, scoring based, deep models

Combined tools

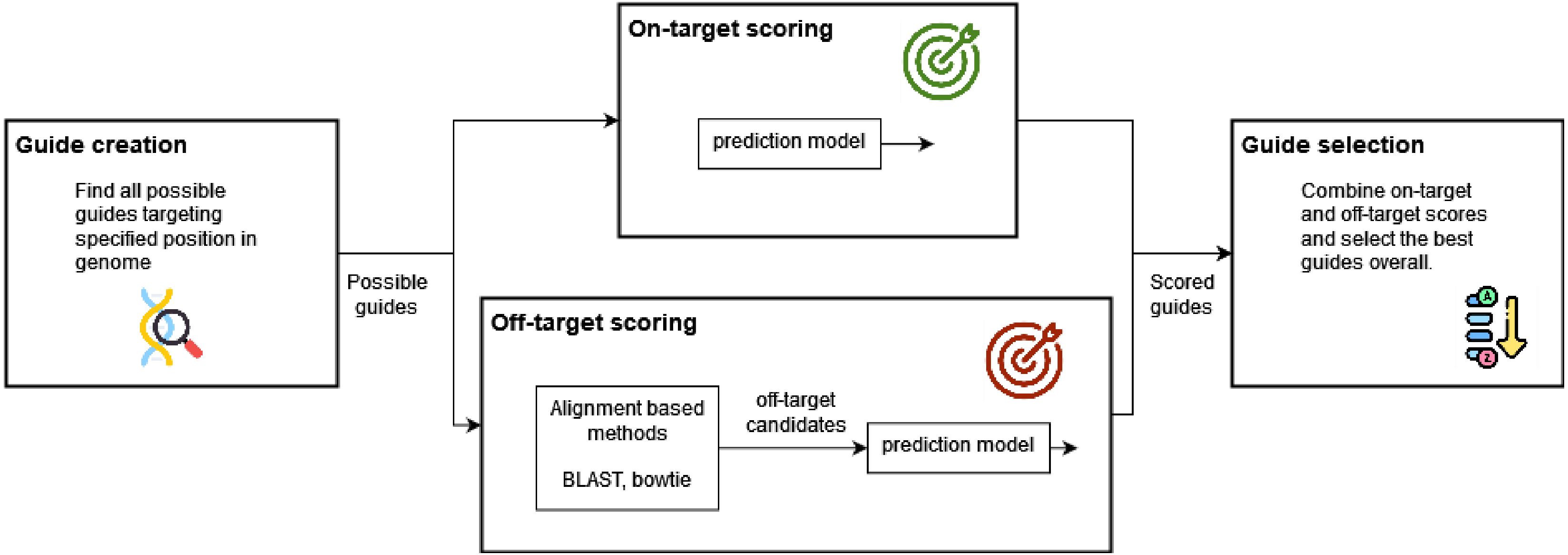
- CHOPCHOP
- DeepCRISPR
- uCRISPR
- Synthego

⚠

- trained on very specific data
- bad documentation
- not all open source
- not reproducible, nor generalizable
- almost no combined ranking

mentioned tools all analyzed in [3]

Project goal: panCRISPR tool



Guide creation

Find all possible guides targeting specified position in genome

Possible guides

On-target scoring

prediction model

Off-target scoring

Alignment based methods

BLAST, bowtie

off-target candidates

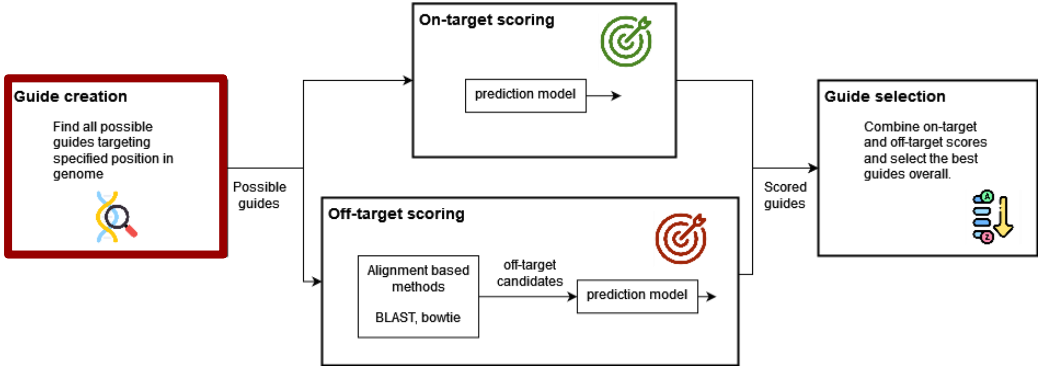
prediction model

Scored guides

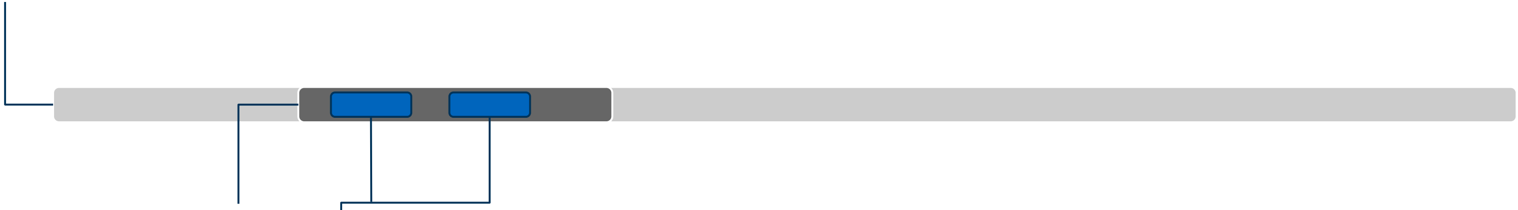
Guide selection

Combine on-target and off-target scores and select the best guides overall.

Guide creation

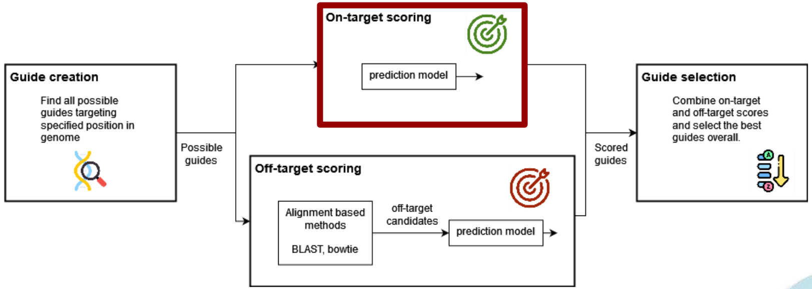


1. user specifies genome and genes
1. download genome file

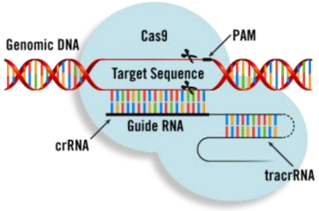


1. identify targets (gene)
2. compute possible guides (20 base pairs)

On-target module

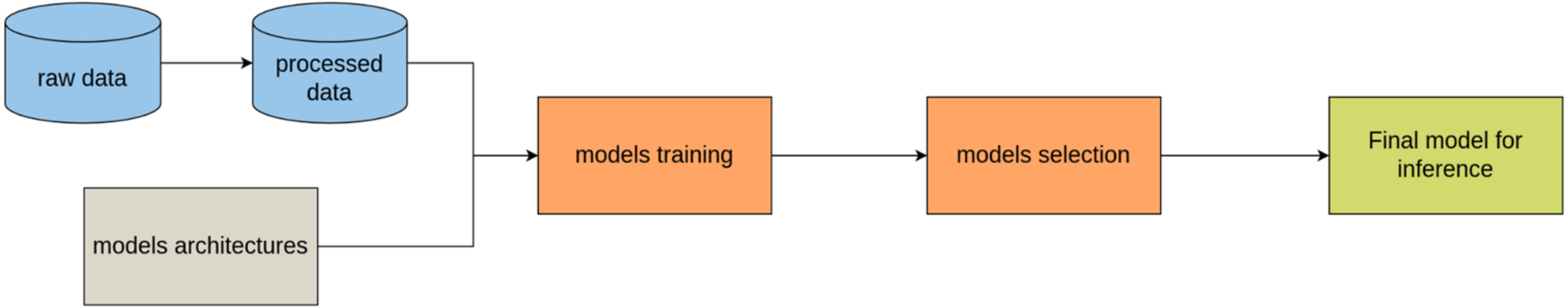


Problem: determine how well a guide RNA bounds to its target (efficiency)



- In-vitro approaches use complex experiments which tend to be expensive
- little is known on what makes a guide efficient

→ predict the efficiency of the guides with a learning algorithm



Data

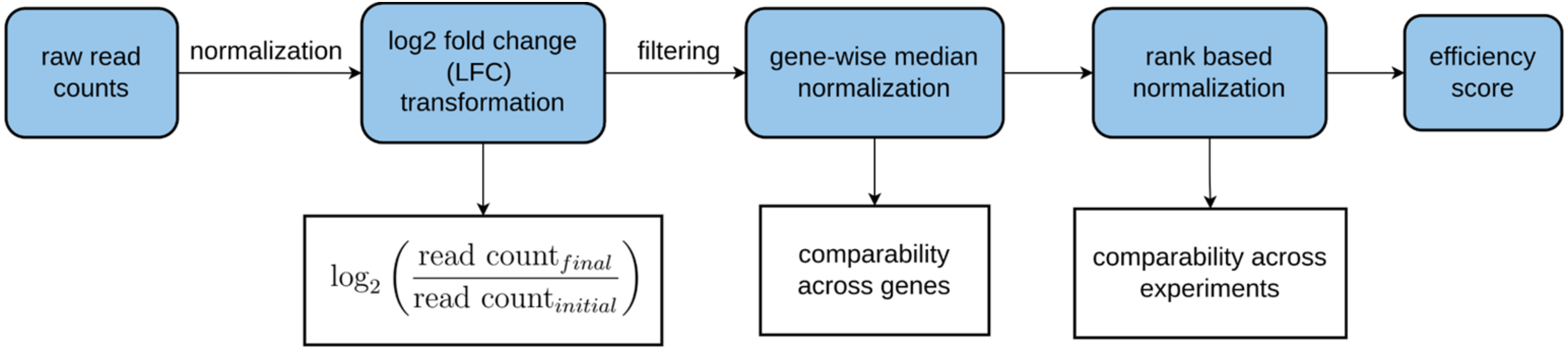
Challenges: ⟶ Few open source data-sets available

⟶ Data comes from different experiments and is difficult to combine

We used the data-sets coming from 3 different experiments (7 cell lines in total)

contains sequences, gene and initial and final read counts

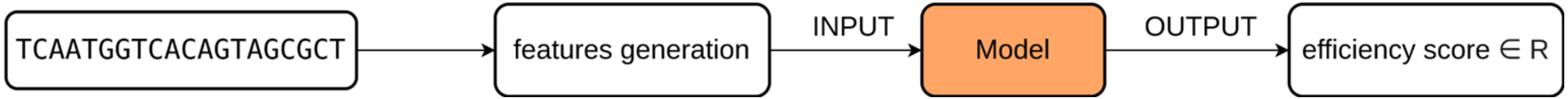
represent the abundance of the correspondent gRNA



raw read counts
normalization
log2 fold change (LFC) transformation
filtering
gene-wise median normalization
rank based normalization
efficiency score
comparability across genes
comparability across experiments

$$\log_2\left(\frac{\text{read count}_{final}}{\text{read count}_{initial}}\right)$$

Models



Shallow model: tree based Gradient Boosting Regressor (GBR)

Features generated from the sequences:

- positional features: occurrence in the sequence of n adjacent nucleotides (G or AC)
- gap features: how often 2 nucleotides appear at a certain distance (A _ _ _ _ _ C)
- biological features: GC content and gRNA melting temperature, defined key features in [2]

[2] Xi Xiang et al. “Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning”. In: Nature communications 12.3238 (2021)

Deep models

INPUT: one-hot encoding of the sequence (1D image with 4 channels)

- baseline_nn: fully connected network with 2 hidden layers and leaky ReLu activations
- CRISPRon: convolutional layers with filters of 3 diff sizes, output flattened and fed into baseline_nn, based on the architecture presented in [2].

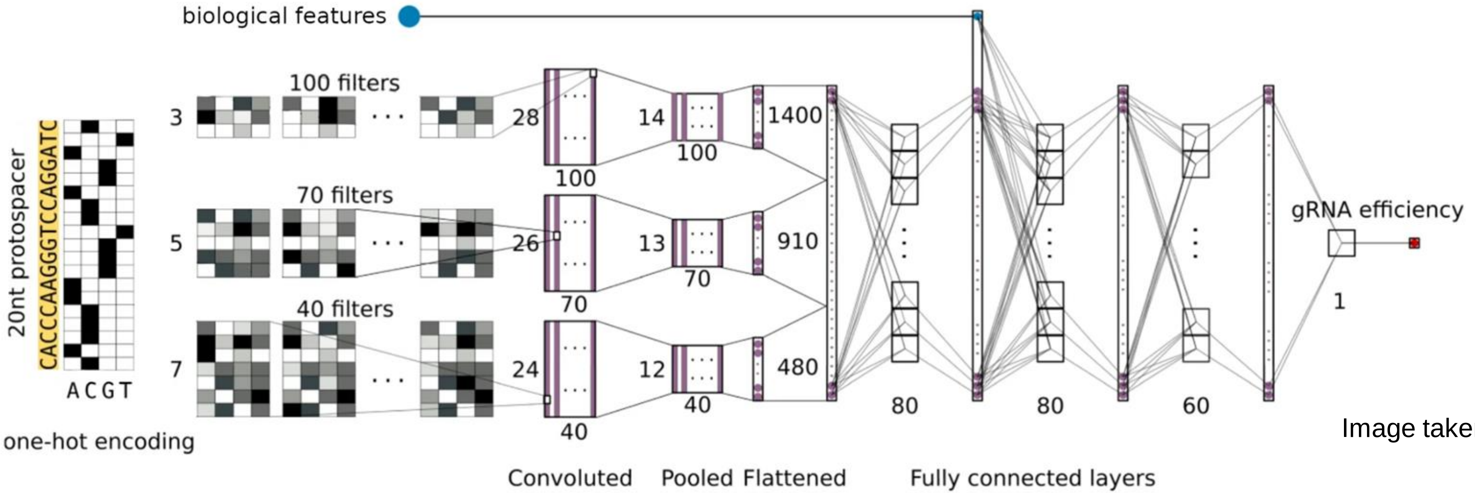
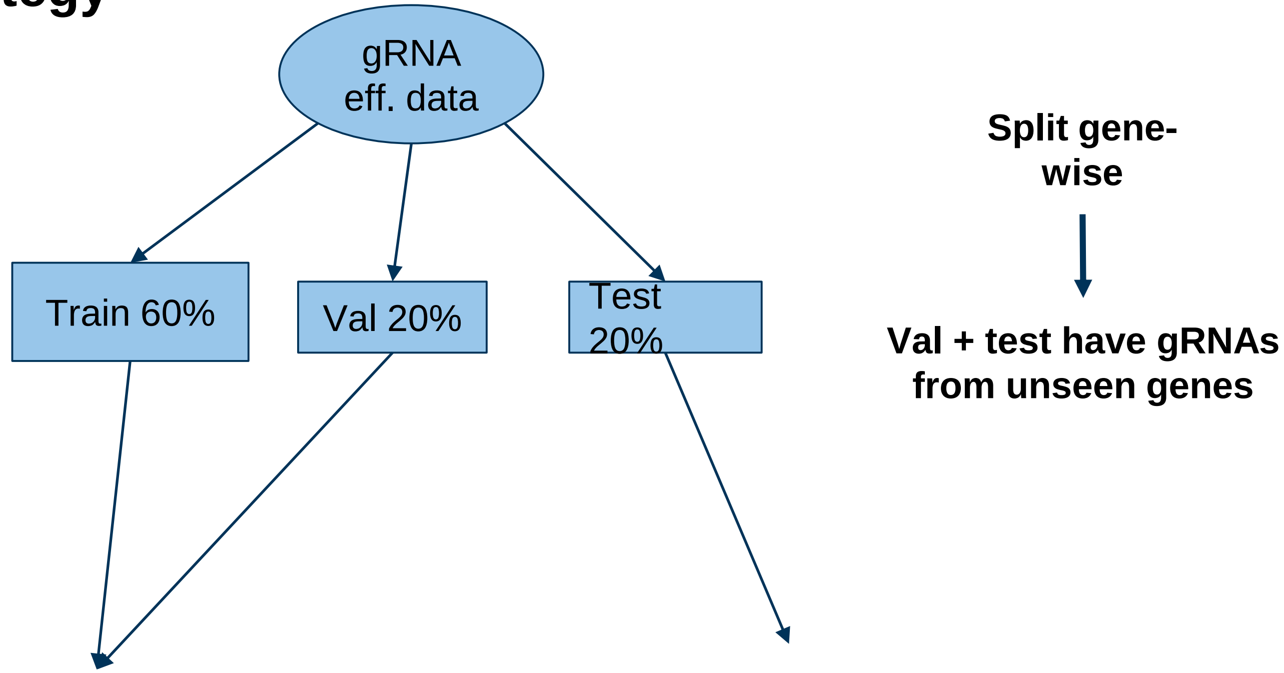


Image taken from [2]

[2] Xi Xiang et al. "Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning". In: Nature communications 12.3238 (2021)

Training strategy



gRNA eff. data

Train 60%

Val 20%

Test 20%

Split gene-wise

Val + test have gRNAs from unseen genes

Train and validate models:

- Hyperparameter tuning (MSE)
- Prevent overfitting (Early stopping)

Evaluate models:

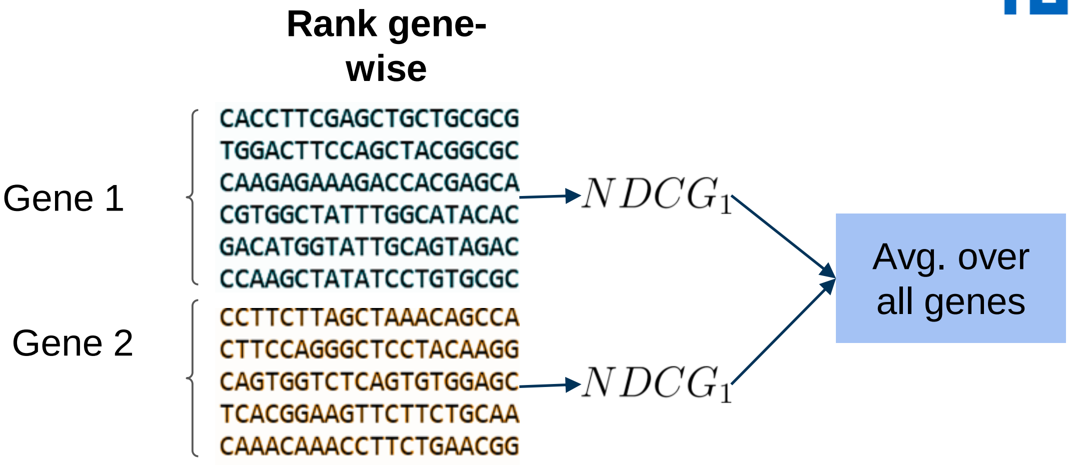
- What evaluation metric to use?

Evaluation metric

$$NDCG_p = \frac{DCG_p}{IDCG_p} \in [0, 1]$$

$$DCG_p = \sum_{i=1}^{p} \frac{eff_i}{log_2(i+1)}$$

$$IDCG_p = \sum_{|rel_p|} \frac{eff_i}{log_2(i+1)}$$



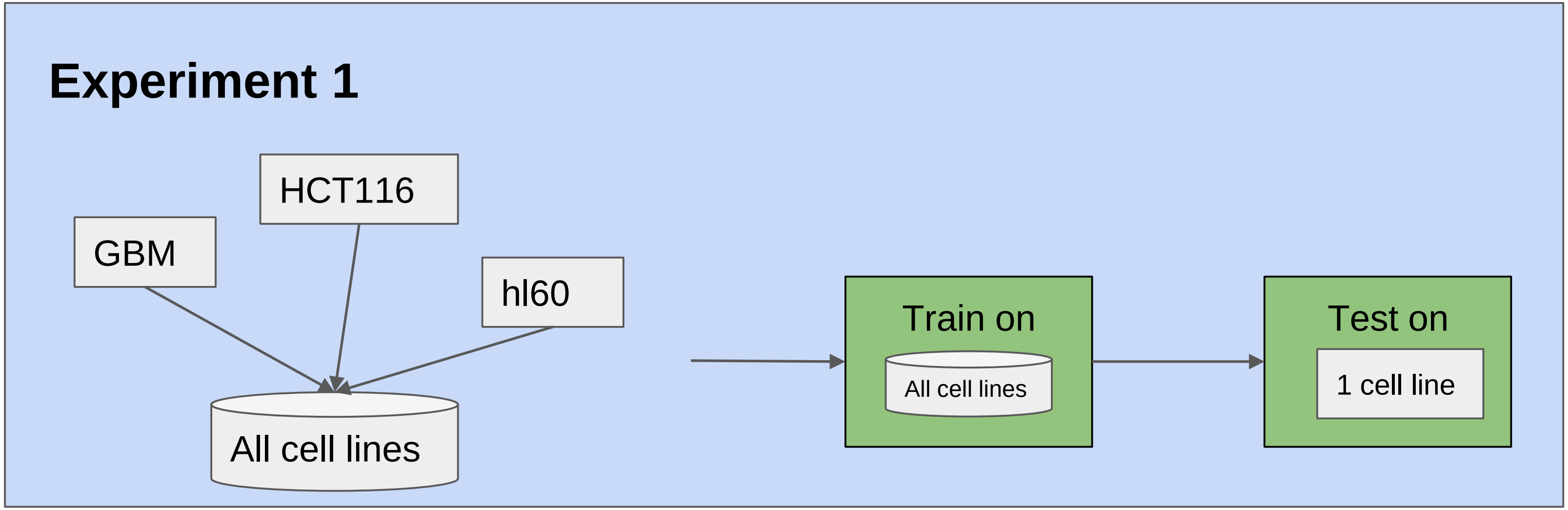
CACCTTCGAGCTGCTGCGCG
TGGACTTCCAGCTACGGCGC
CAAGAGAAAGACCACGAGCA
CGTGGCTATTTGGCATACAC
GACATGGTATTGCAGTAGAC
CCAAGCTATATCCTGTGCGC

CCTTCTTAGCTAAACAGCCA
CTTCCAGGGCTCCTACAAGG
CAGTGGTCTCAGTGTGGAGC
TCACGGAAGTTCTTCTGCAA
CAAACAAACCTTCTGAACGG

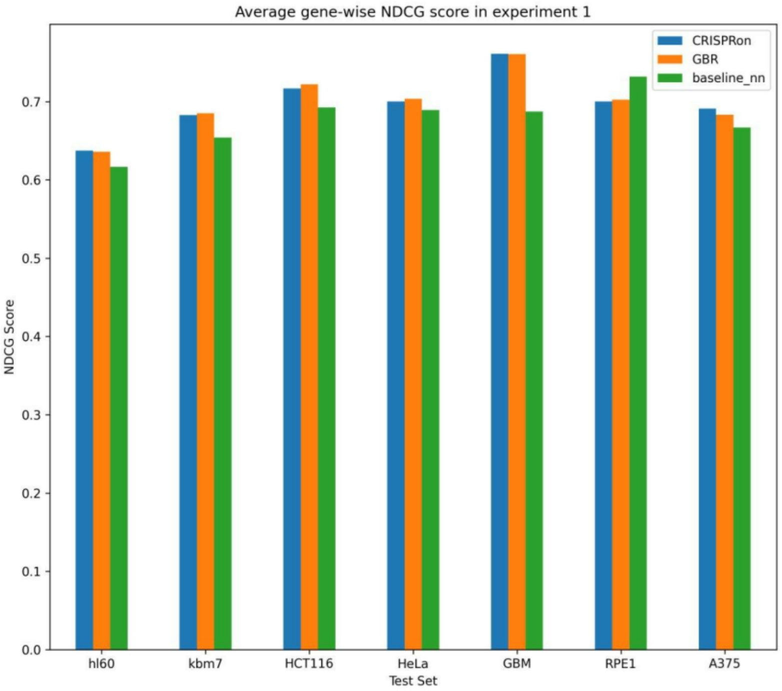
✓ Consider p most efficient guides

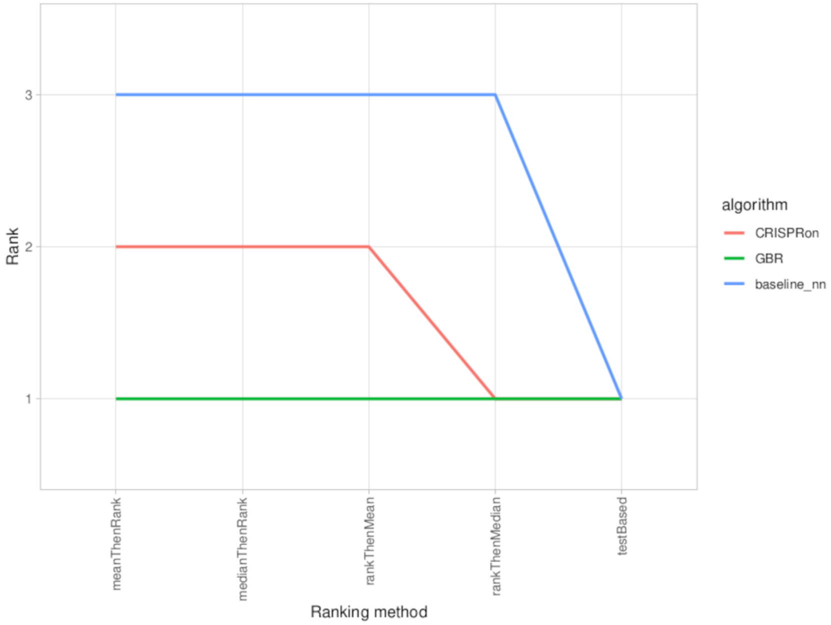
✓ Penalize model for ranking highly efficient guides poorly

Results


Experiment 1
HCT116
GBM
hl60
All cell lines
Train on
All cell lines
Test on
1 cell line

Results: Experiment 1



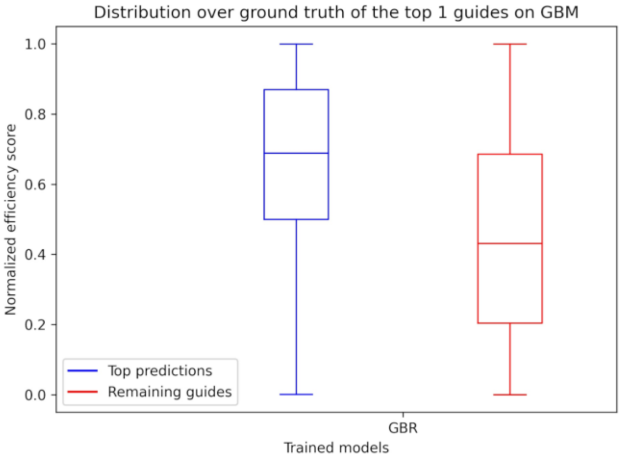


Line graph created using ChallengeR toolbox [1]

[1] Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results.

Results: Experiment 1

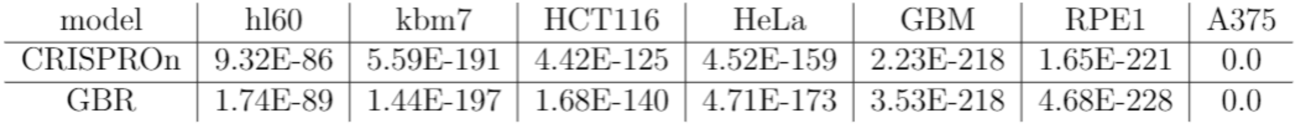
Do our models detect the best guides?



Is the distributional shift significant?

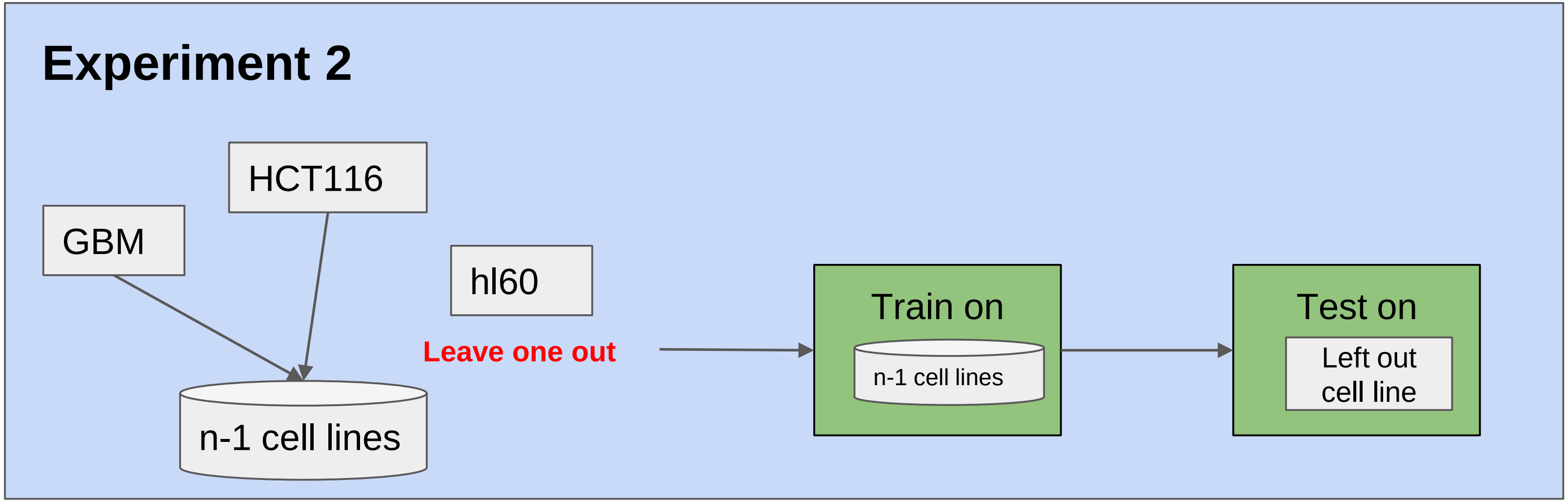
✓

P-values for Wilcoxon test:

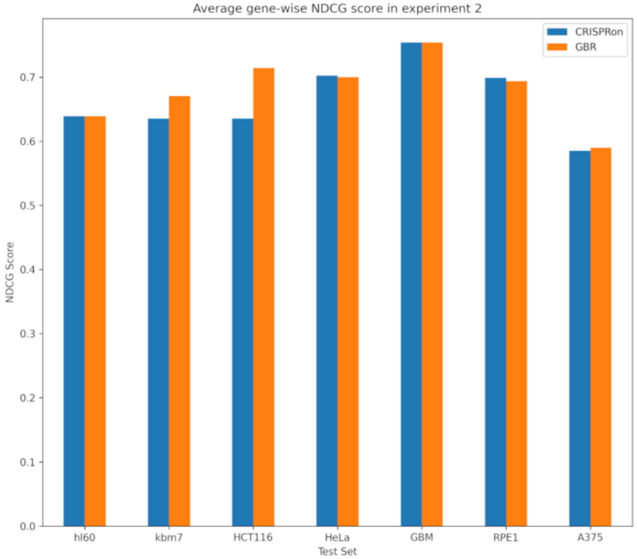


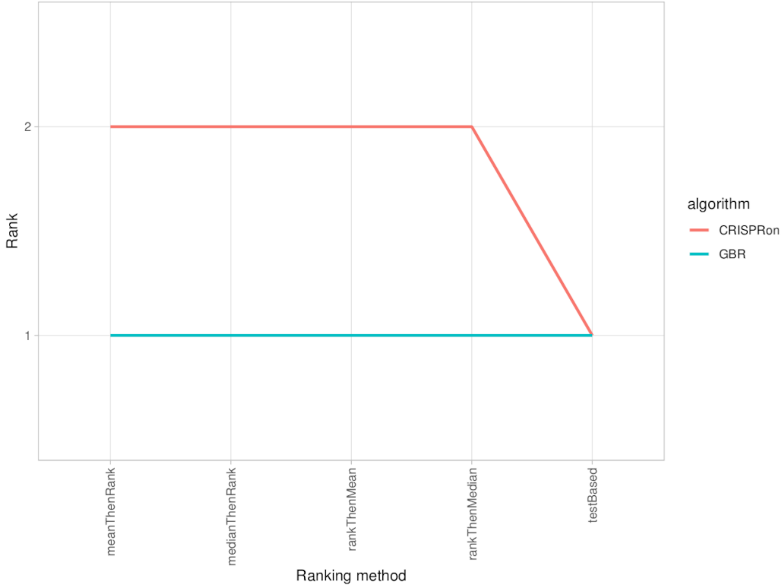
model	hl60	kbm7	HCT116	HeLa	GBM	RPE1	A375
CRISPROn	9.32E-86	5.59E-191	4.42E-125	4.52E-159	2.23E-218	1.65E-221	0.0
GBR	1.74E-89	1.44E-197	1.68E-140	4.71E-173	3.53E-218	4.68E-228	0.0

Results


Experiment 2
HCT116
GBM
hl60
Leave one out
n-1 cell lines
Train on
n-1 cell lines
Test on
Left out cell line

Results: Experiment 2

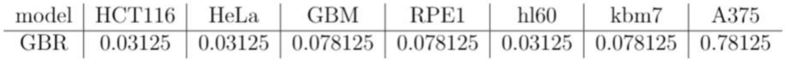




Line graph created using ChallengeR toolbox [1]

Does combining cell lines give better transferability? ✓

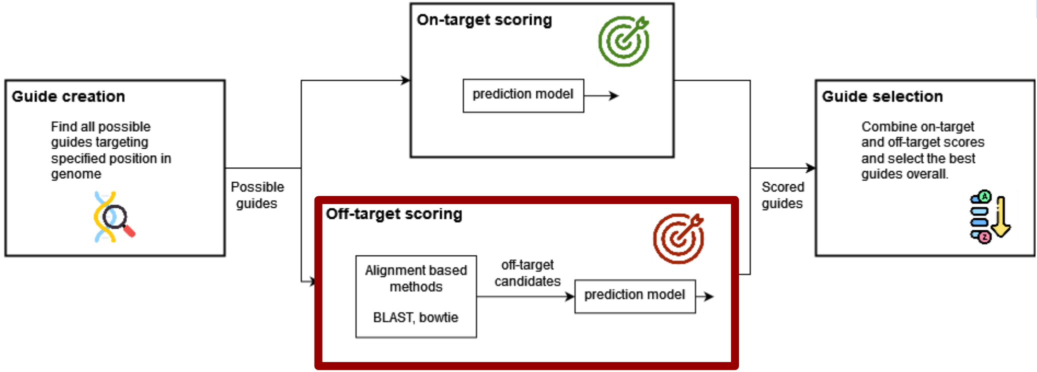
P-values for Wilcoxon test:



model	HCT116	HeLa	GBM	RPE1	hl60	kbm7	A375
GBR	0.03125	0.03125	0.078125	0.078125	0.03125	0.078125	0.78125

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. [1]

Off-target module



Guide creation

Find all possible guides targeting specified position in genome

Possible guides

On-target scoring

prediction model

Off-target scoring

Alignment based methods

BLAST, bowtie

off-target candidates

prediction model

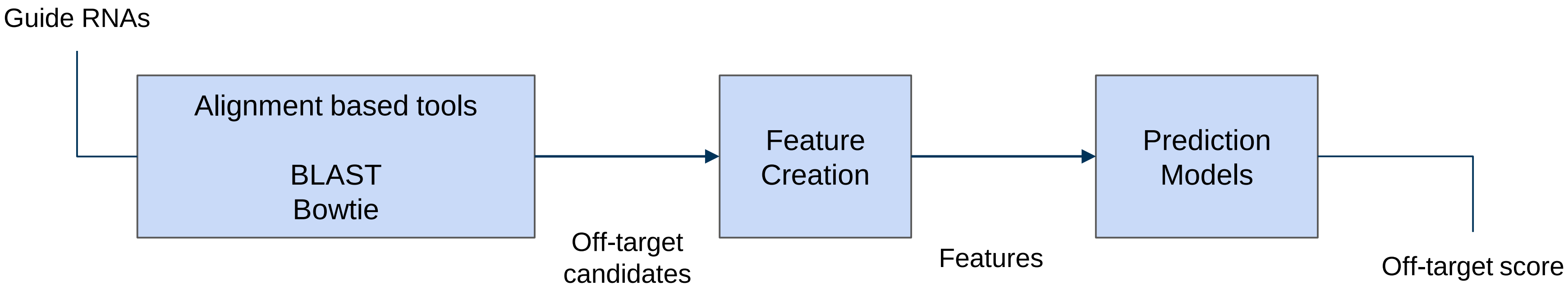
Scored guides

Guide selection

Combine on-target and off-target scores and select the best guides overall.

Problem: determine how many off-targets a guide RNA can have (specificity)

⇨ Predict off-targets and score them



Guide RNAs

Alignment based tools

BLAST
Bowtie

Off-target candidates

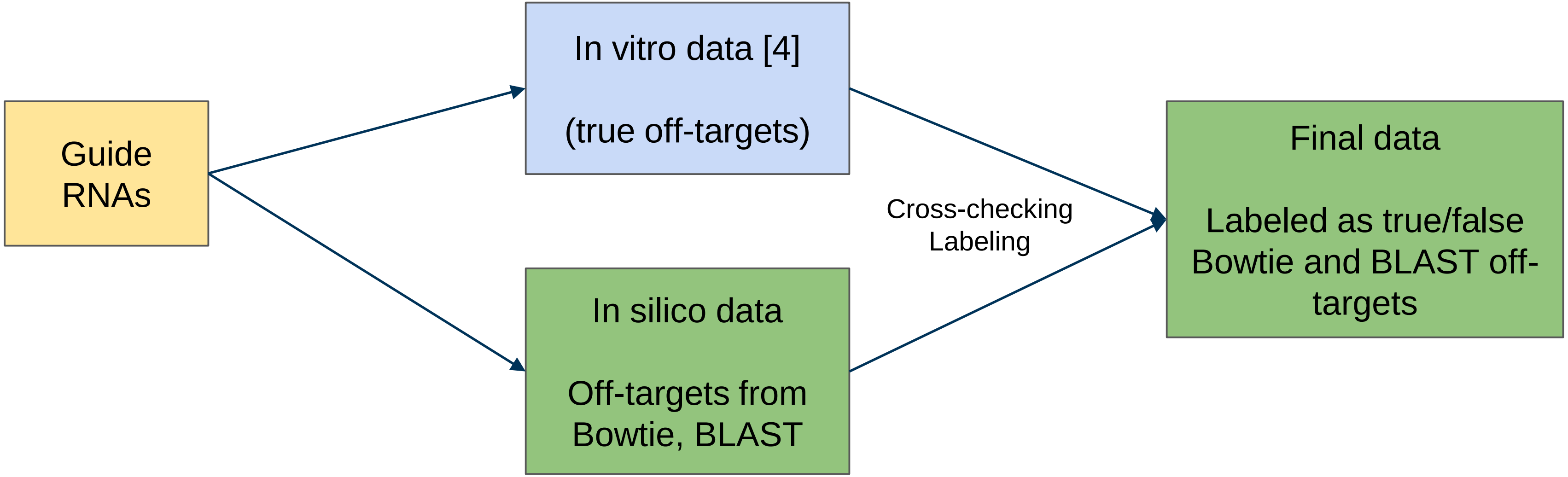
Feature Creation

Features

Prediction Models

Off-target score

Data



Guide RNAs

In vitro data [4]

(true off-targets)

In silico data

Off-targets from Bowtie, BLAST

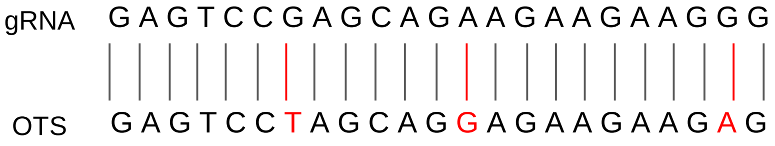
Cross-checking Labeling

Final data

Labeled as true/false Bowtie and BLAST off-targets

[4] Jifang Yan et al. “Benchmarking and integrating genome-wide CRISPR off-target detection and prediction”. In: Nucleic Acids Research 48.20 (Nov. 2020), pp. 11370–11379.

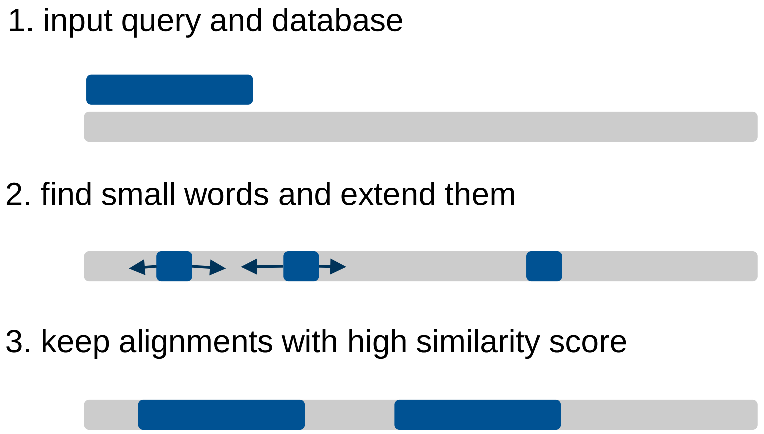
Bowtie



```
gRNA  G A G T C C G A G C A G A A G A A G A A G G G
      | | | | | | | | | | | | | | | | | | | | | | |
OTS   G A G T C C T A G C A G G A G A A G A A G A G
```

- End-to-end alignment (whole gRNA sequence)
- Finds OTS with up to 3 mismatches

BLAST



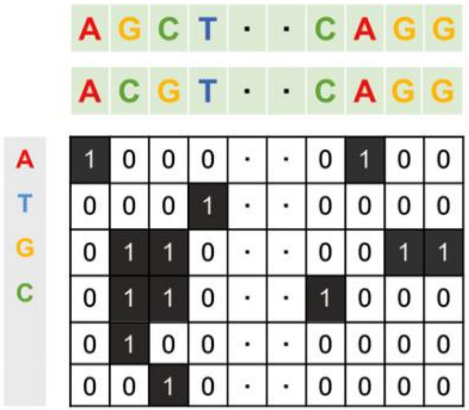
1. input query and database
2. find small words and extend them
3. keep alignments with high similarity score

- Local alignment tool (some part of gRNA)
- Doesn’t have a restriction on mismatches
- Finds alignments based on evolutionary similarity

Features

gRNA	GAGTCCGAGCAGAAGAAGAAGGG	(23 nt long sequence)	Bowtie, BLAST
OTS	GAGTCCTAGCAGGAGAAGAAGAG	(23 nt long sequence)	
GC-content	GAGTCCGAGCAGAAGAAGAAGGG	(% of G, C)	Biological features
Melting temperature	temperature to cause double strand break	(in °C)	

Encoding

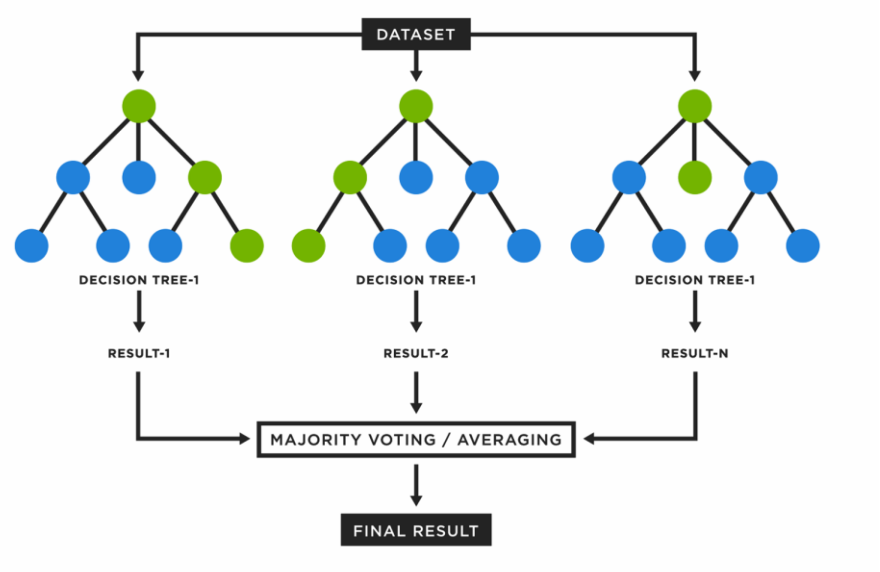


gRNA	A	G	C	T	·	·	C	A	G	G
OTS	A	C	G	T	·	·	C	A	G	G
A	1	0	0	0	·	·	0	1	0	0
T	0	0	0	1	·	·	0	0	0	0
G	0	1	1	0	·	·	0	0	1	1
C	0	1	1	0	·	·	1	0	0	0
	0	1	0	0	·	·	0	0	0	0
	0	0	1	0	·	·	0	0	0	0

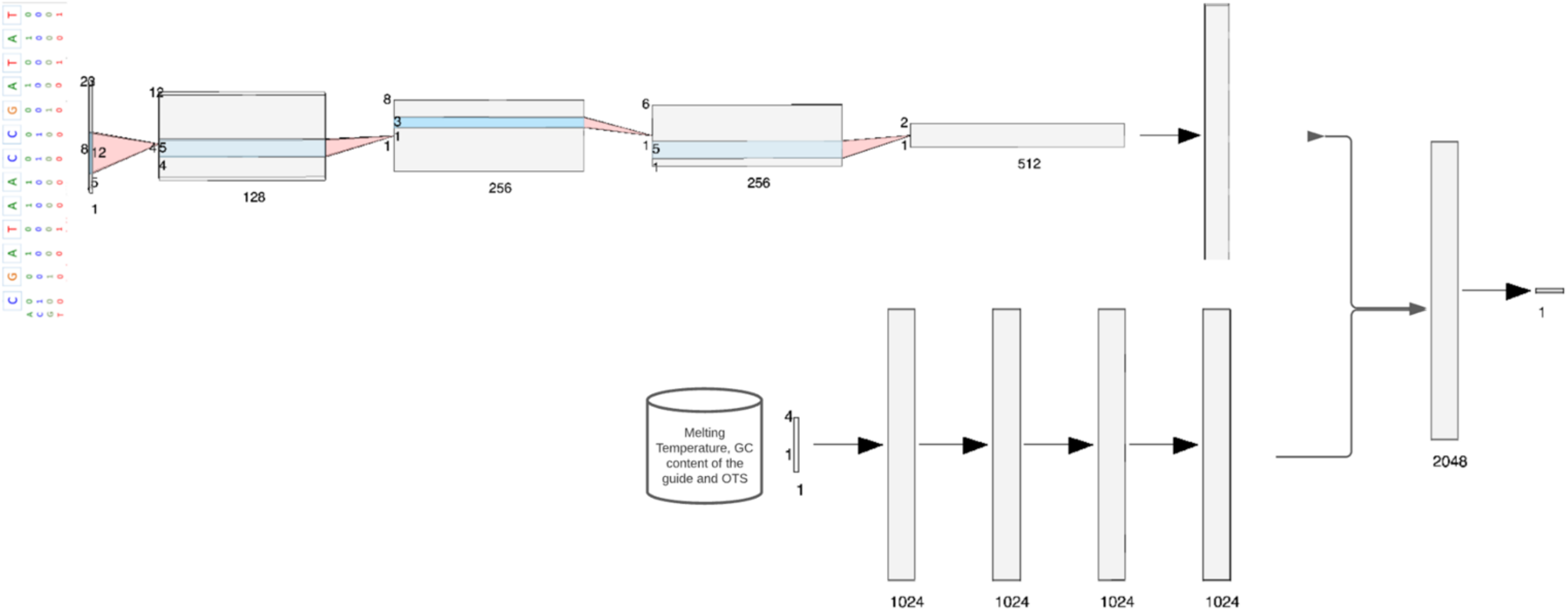
Figure 10: 6-bit encoding scheme

Image from [5] Jiecong Lin et al. “CRISPR-Net: A Recurrent Convolutional Network Quantifies CRISPR Off-Target Activities with Mismatches and Indels”. In: Advanced science 7.13 (2020).

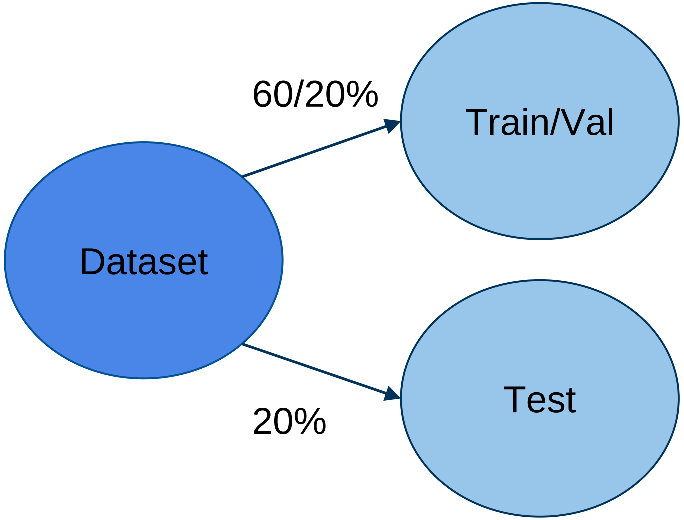
Shallow models: random forest



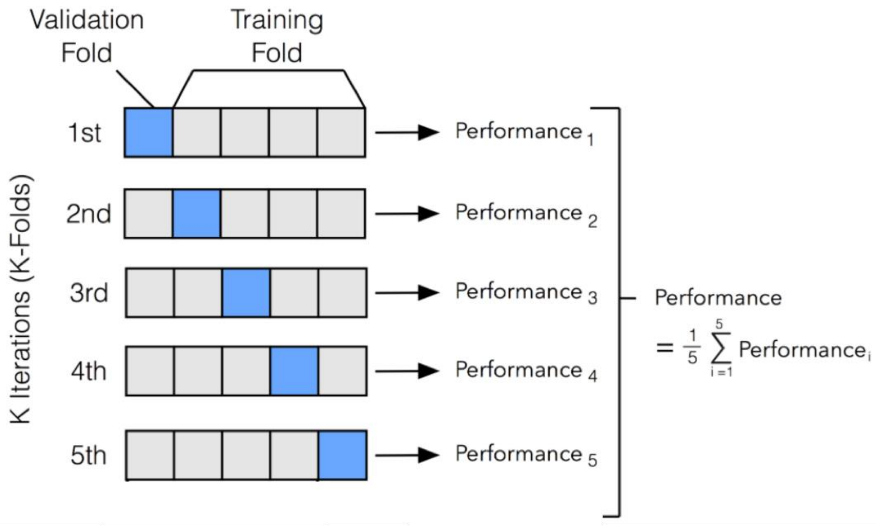
Models: deep convolutional nn and random forest



Training strategy



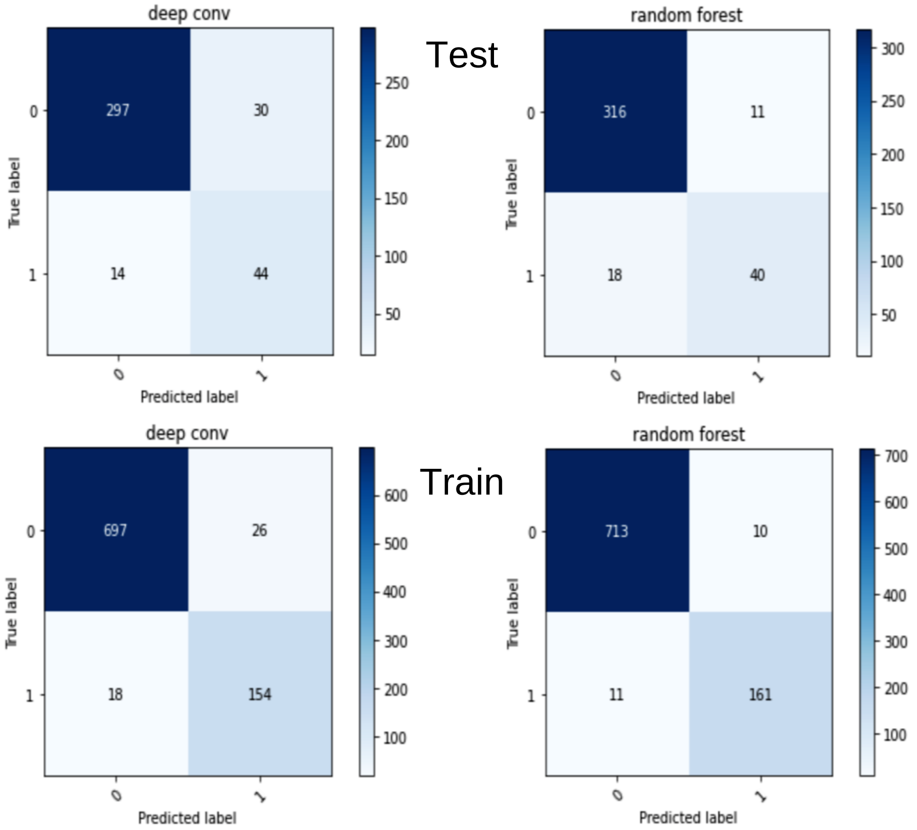
Test on different guides - test generalization



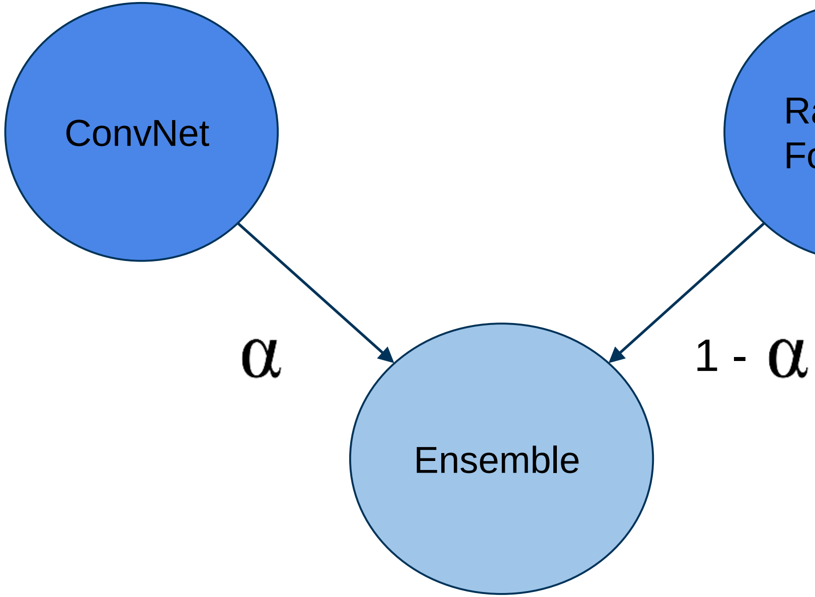
Each split puts a single guide and associated ots into the validation fold and train on the remaining

Evaluation

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

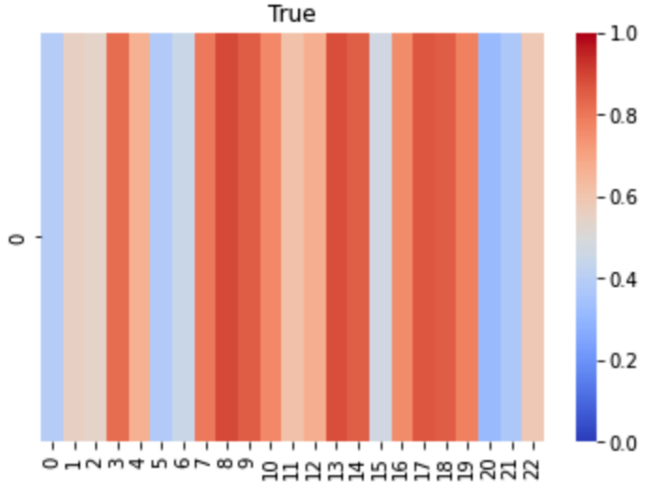


Model ensemble: combine models by weightening the results

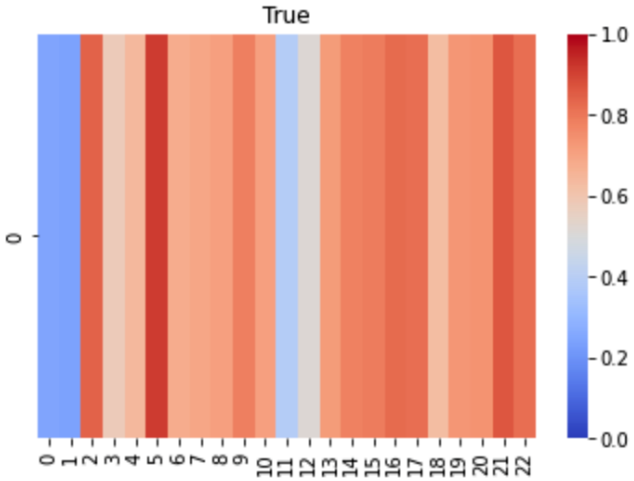


Results:

Occlusion of the nucleotides asserts that end proximal regions have a direct effect on the probability of being an OTS

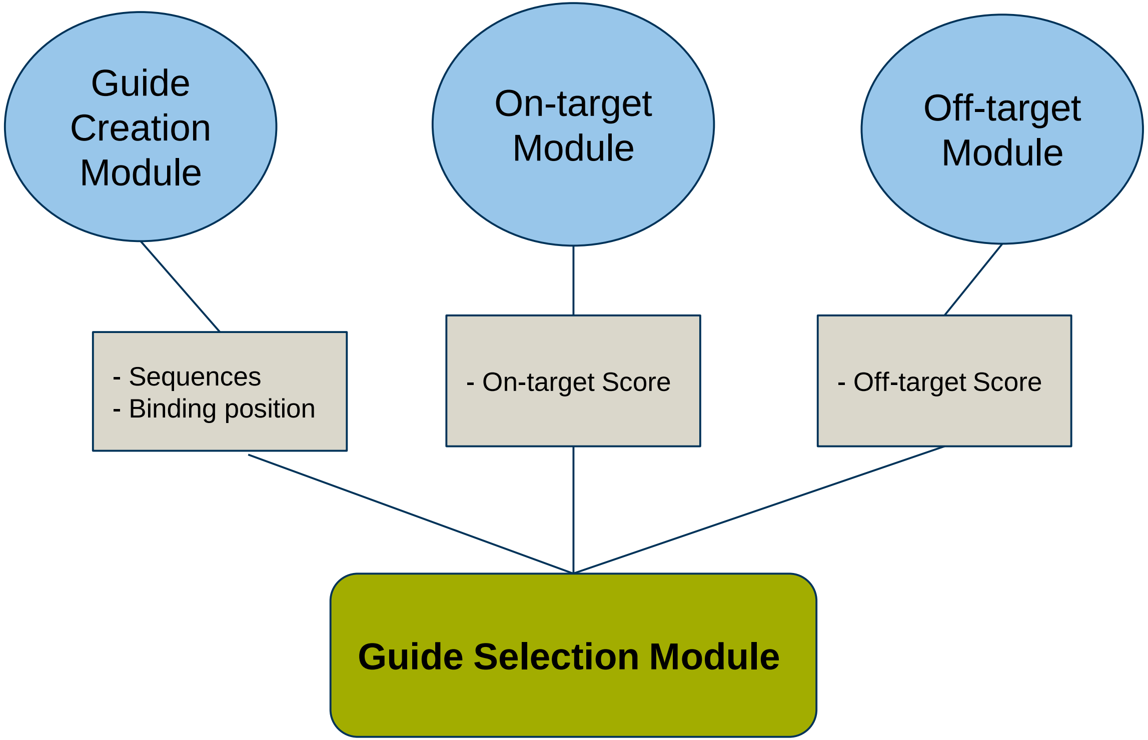


nucleotide position



nucleotide position

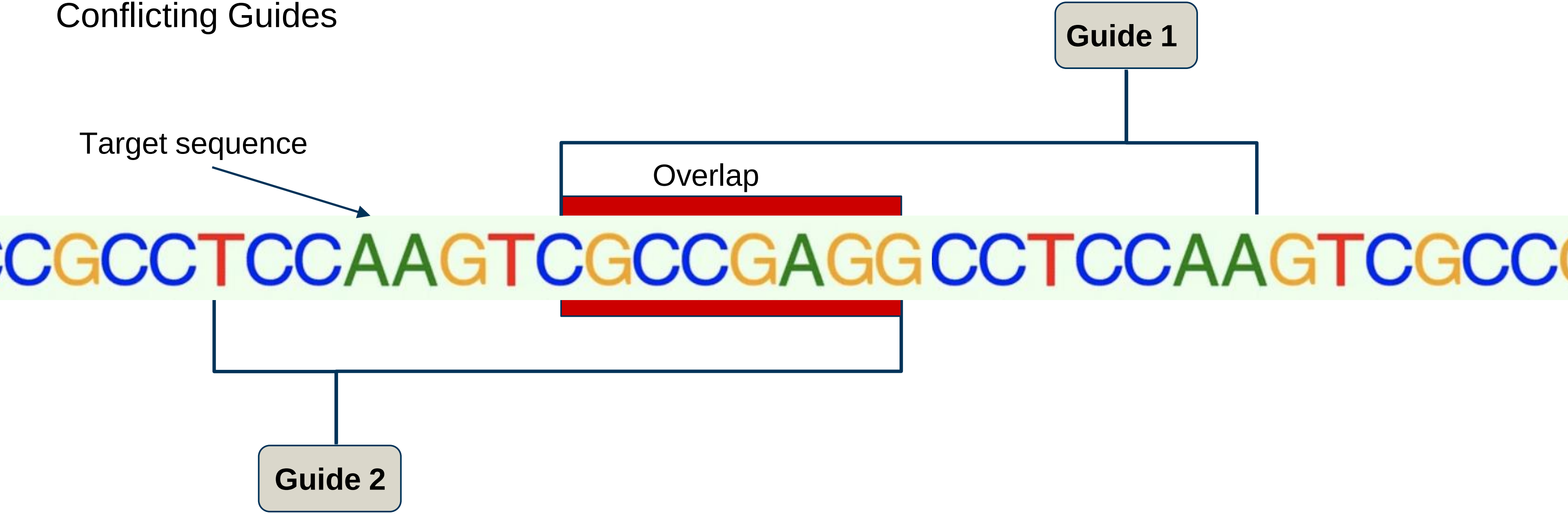
Guide selection module



GOAL: Select guides with higher On-target activity and lower Off-target activity

Guide selection module

Conflicting Guides



Guide 1 and **Guide 2** will compete over the same binding sub-sequence

Only one of them could eventually bind to the target sequence

Guide selection module

Case 1: Allow overlaps

Type of experiments:

Lentiviral vectors based experiments where on average only one guide is delivered into the cell.

Approach:

1. Set ON and OFF default (0.5, 0.5)
2. Compute overall score as weighted sum:

$$SCORE = ON * SCORE_{ON} + OFF * SCORE_{OFF}$$

1. Select the P highest scoring guides

Guide selection module

Case 2: Penalize overlaps

Type of experiments:

Multiplexed experiments where multiple guides are delivered into cells.

Objective function:

$$x^* = \arg \max_{x=[x_k]_{k=1..n}} x^T * SCORE - \lambda x^T M x$$

$$\text{subject to } \Sigma_i x_i = p$$

$M = [M_{i,j}]_{1 \leq i,j \leq n}$ $\quad M_{i,j} = \begin{cases} 1 & \text{if } i^{th} \text{ and } j^{th} \text{ guides are conflicting} \\ 0 & \text{else} \end{cases}$

p Number of selected guides

We used “qubovert“** python package for Polynomial Constrained Boolean Optimization

**https://github.com/jtiosue/qubovert

Guide selection module

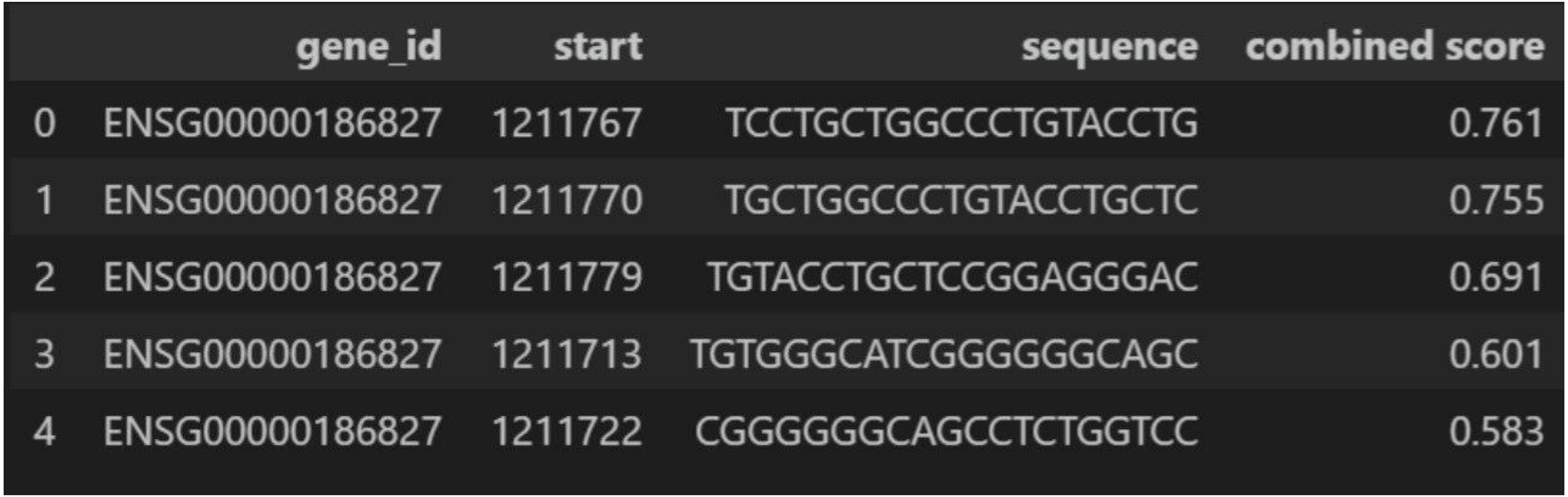
Case 2: Penalize overlaps

Greedy approach:

```
1. Set ON and OFF, default (0.5, 0.5)
2. Compute overall score for every guide
3. Repeat p times:
    1. Select the highest scoring guide
    2. discard all conflicting guides from the pool
```

- Fast approach
- Discard all conflicts in the selected guides
- Provide sub-optimal solution

Framework output



	gene_id	start	sequence	combined score
0	ENSG00000186827	1211767	TCCTGCTGGCCCTGTACCTG	0.761
1	ENSG00000186827	1211770	TGCTGGCCCTGTACCTGCTC	0.755
2	ENSG00000186827	1211779	TGTACCTGCTCCGGAGGGAC	0.691
3	ENSG00000186827	1211713	TGTGGGCATCGGGGGGCAGC	0.601
4	ENSG00000186827	1211722	CGGGGGGCAGCCTCTGGTCC	0.583

Conclusion

- Our solution provides a complete framework for panCRISPR experiments:
  - Novelty: On and OFF target assessment in one end to end solution

- Our models performances (On/Off -target) show comparable performances to the state of the art.

- We have introduced new evaluation metrics that haven’t been used in the literature.

- Our guide selection module covers different types of experiments.

- We have produced clean, modular, and extensible code as a good basis for further improvement.

Outlook and Discussion

- Further performance improvement can be brought by adding more datasets (further cell lines, genes, …)

- Potential improvement with better and more complex featurization

- Runtime of our toolkit can be improved.

References

[1] Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. Sci Rep 11, 2369 (2021). https://doi.org/10.1038/s41598-021-82017-6

[2] Xi Xiang et al. “Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning”. In: Nature communications 12.3238 (2021)

[3] Guanqing Liu, Yong Zhang, Tao Zhang,
Computational approaches for effective CRISPR guide RNA design and evaluation, 2020
https://www.sciencedirect.com/science/article/pii/S2001037019303551

[4]

[5] Jiecong Lin et al. “CRISPR-Net: A Recurrent Convolutional Network Quantifies CRISPR Off-Target Activities with Mismatches and Indels”. In: Advanced science 7.13 (2020).