Kipoi: Accelerating the Community Exchange and Reuse of Predictive Models for Genomics

Žiga Avsec[1,2], Roman Kreuzhuber[3,4], Johnny Israeli[5], Nancy Xu[5], Jun Cheng[1,2], Avanti Shrikumar[5], Abhimanyu Banerjee[5], Daniel S. Kim[5], Lara Urban[4,6], Anshul Kundaje[5], Oliver Stegle[4,6], Julien Gagneur[1]

1 Technical University Munich
2 QBM Graduate School, Ludwig-Maximilians Universität, Munich
3 Department of Haematology, University of Cambridge
4 European Molecular Biology Laboratory, European Bioinformatics Institute
5 Stanford University
6 European Molecular Biology Laboratory, Genome Biology Unit

Reviewed by: Brandon Liu

https://qdata.github.io/deep2Read/

Outline

Introduction

Approach

Experiments

Discussion

Future Work

References

Introduction

- Predictive ML models have widespread usage in genomics.
- Despite importance of these models, it is very difficult to share and exchange models effectively.
- No established standard for sharing **trained** models.
- Challenge: heterogeneity of genomics technologies, techniques and frameworks, many specific data pre-processing strategies, and ease-of use for practitioners not expert in machine learning
- What: API and repository of ready-to-use genomics models.
- Goal: foster the dissemination and use of machine learning models in genomics.

Outline

Introduction

Approach

Experiments

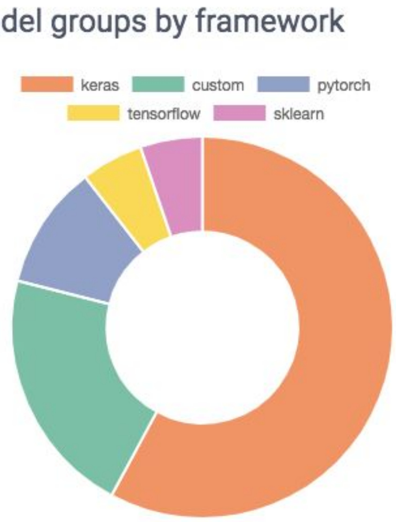
Discussion

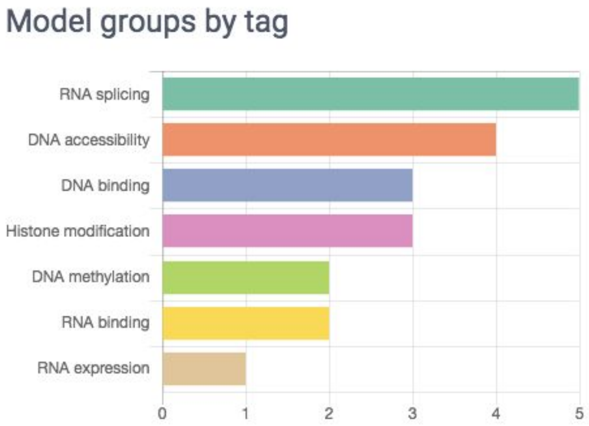
Future Work

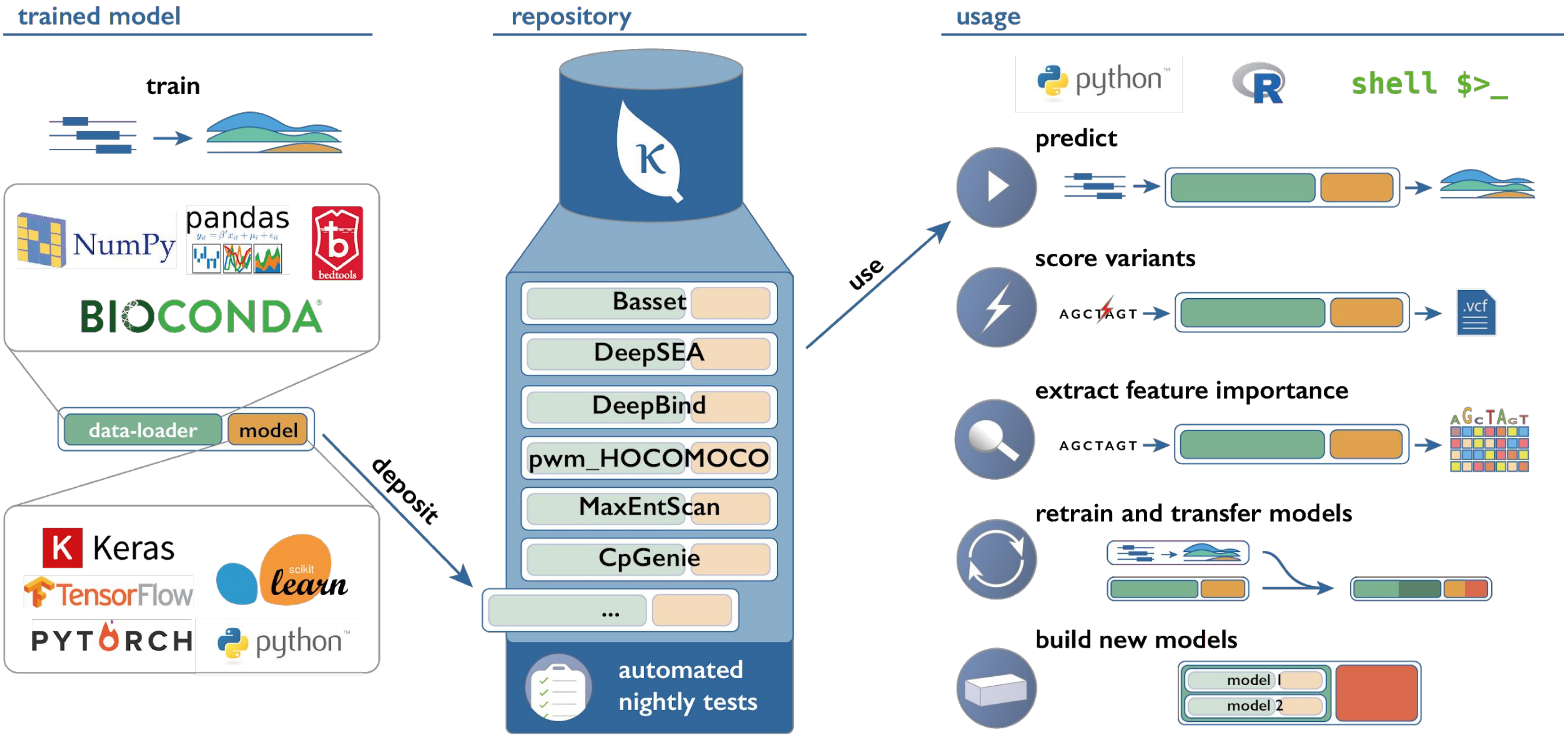
References

Approach

- Standardized data handling (data-loaders) for genomic data types
- 2000 trained models on Github
- API for accessing models


Model groups by framework


Model groups by tag



trained model
train
NumPy
pandas
bedtools
BIOCONDA
data-loader
model
Keras
TensorFlow
scikit learn
PYTORCH
python
deposit
repository
K
Basset
DeepSEA
DeepBind
pwm_HOCOMOCO
MaxEntScan
CpGenie
...
automated nightly tests
use
usage
python
R
shell $>_
predict
score variants
AGCTAGT
.vcf
extract feature importance
AGCTAGT
retrain and transfer models
build new models
model 1
model 2

Outline

Introduction

Approach

Experiments

Discussion

Future Work

References

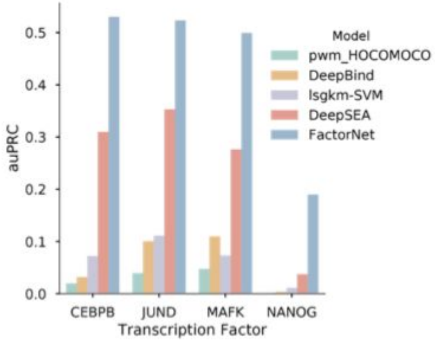
Benchmarking of Alternative Models Predicting Transcription Factor Binding

- Different modeling paradigms, including methods based on classical position weight matrices (PWM), gapped k-mer support vector machines (lsgkm-SVM) and deep learning (DeepBind, DeepSEA, and FactorNet)
- Kipoi model implementations derived from publications, trained by authors, and assessed on chromosome 8 which was not used in training.
- Originally cumbersome task: different software frameworks, different file formats for input, different prediction formats, different software dependencies..

a

Model	Publication	Type	Input modalities	Framework	Size	Prediction time (n=256)	Input seq. length
pwm_HOCOMOCO	Kulakovskiy et al 2015	Position weight matrix scan	DNA sequence	Keras	16KB	1.2ms	101bp
DeepBind	Alipanahi et al 2015	Convolutional neural network	DNA sequence	Keras	36KB	2.4ms	101bp
lsgkm-SVM	Ghandi et al 2014	Support vector machine	DNA sequence	LS-GKM	4MB	10s	101bp
DeepSEA	Zhou *et al 2015*	Convolutional neural network	DNA sequence	PyTorch	200MB (605MB)	94ms	1000bp
FactorNet	Quang *et al* 2017	Convolutional and recurrent neural network	DNA sequence + DNase + annotation	Keras	3-13MB	118ms	1002bp

b



c

```
# Create and activate a new
# conda environment with all
# model dependencies installed
kipoi env create <Model>
source activate kipoi-<Model>

# Run model prediction and save
# the results sequentially into
# an hdf5 file
kipoi predict <Model> \
  --dataloader_args='{
  “intervals_file”: “intervals.bed”,
  “fasta_file”: “hg38.fa”}' \
  -o ‘<Model>.preds.h5’
```

Figure 2 | Applying and benchmarking alternative Kipoi models for transcription factor binding prediction. (a) Five models for predicting transcription factor binding that are based on alternative modeling paradigms: i) predefined position weight matrices contained in the HOCOMOCO database[23]; ii) lsgkm-SVM[24], a support vector machine classifier; iii) the convolutional neural network DeepBind[5]; iv) the multi-task convolutional neural network DeepSEA; v) FactorNet, a multimodal deep neural network with convolutional and recurrent layers that further integrates chromatin accessibility profile and genomic annotation features. Models differ by i) the size of genomic input sequence, where DeepSEA[6] and FactorNET[7] consider ~1 kb sequence inputs, whereas other models are based on ~100 bp, and ii) parametrization complexity with the total size of model parameters ranging from 16kB (pwm_HOCOMOCO) to 200 Mb (DeepSEA). **(b)** Performance of the models in **a** for predicting ChIP-seq peaks of four transcription factors on held-out data (chromosome 8), quantified using the area under the precision-recall curve. More complex models yield more accurate predictions than basic models which are commonly used. **(c)** Example access to Kipoi models via the command line interface to install required software dependencies, download the model, extract and pre-process the data, and write predictions to a new file. Results as shown in **b** can be obtained for all Kipoi models using this generic command. Placeholder <Model> can be any of the models listed in **a**.

Improving Predictive Models of Chromatin Accessibility using Transfer Learning

- Transfer learning for adapting/reusing models for a similar task.
- Enables more rapid training, requires less data to train, and improves predictive performance compared to models trained from scratch.
- Example: edge detection for images or transcription factor motifs in genomics are repeat problems in DNN.
- Started with 431 biosamples, held-out 10, leaving 421 for training a genome-wide model for predicting chromatin accessibility.
- For 10 held-out samples, trained a new model while keeping all but last 2 layers fixed during training.
- Transfer model ~15.2% improvement in area under precision-recall curve compared to model initialized with random parameters.
- 2.8 epoch average vs 17.3 epoch average training improvement.

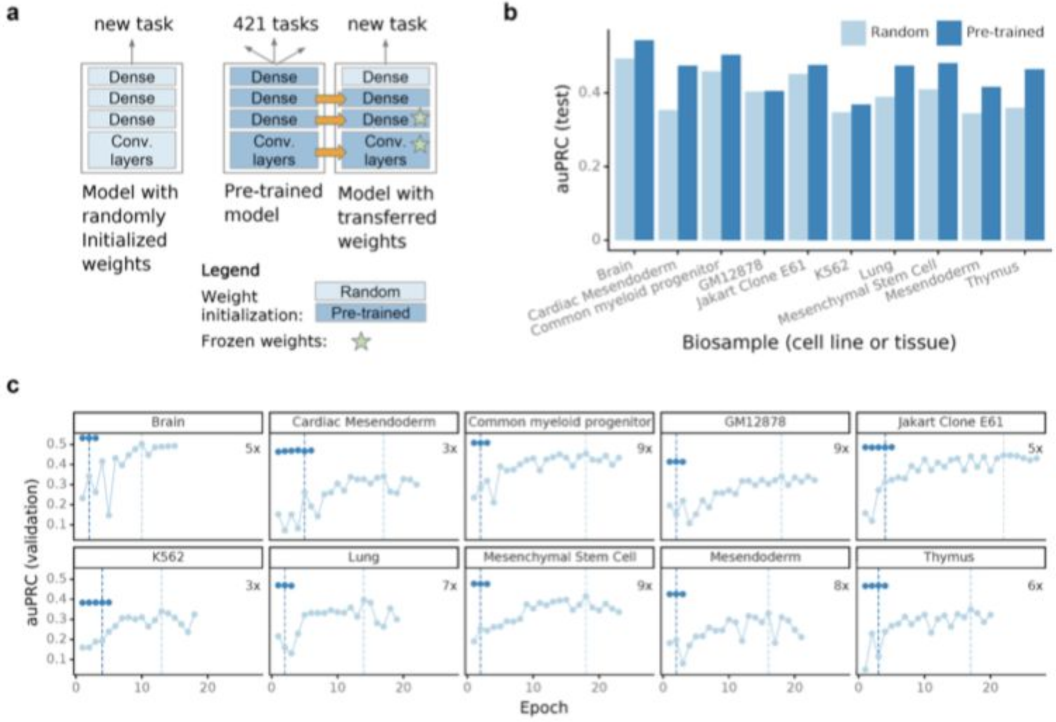
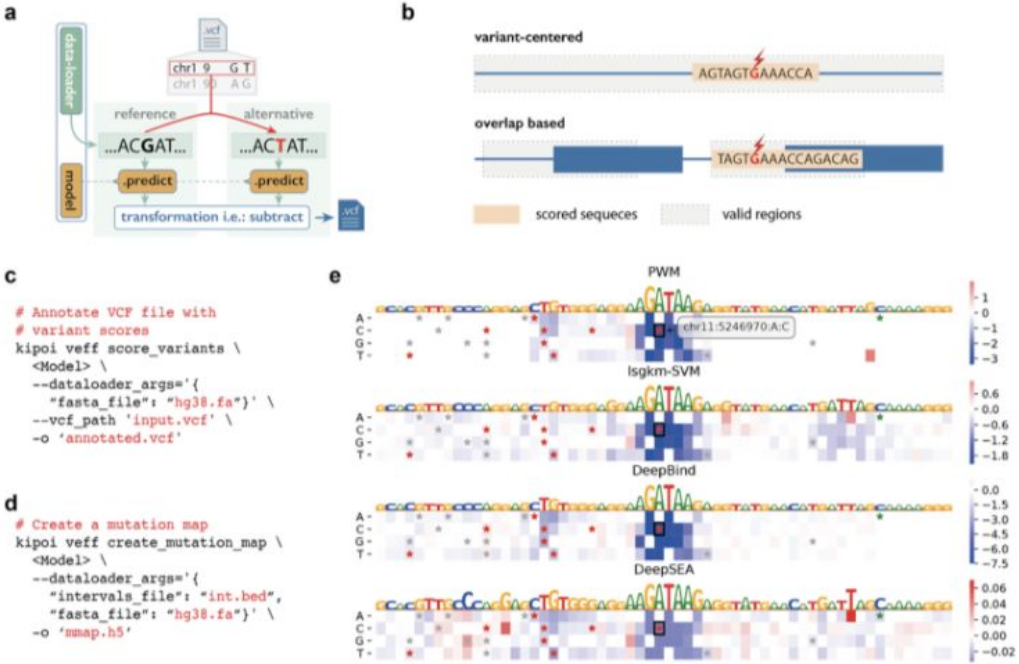


Figure 3: Adapting existing models to new tasks (transfer learning). (a) Architecture of alternative models for predicting chromatin accessibility from DNA sequence. Model parameters are either randomly initialized (left) or transferred from an existing neural network pre-trained on 421 other biosamples (cell lines or tissues, right). (**b)** Prediction accuracy measured using the area under the precision-recall curve, comparing randomly initialized (light blue) versus pre-trained (dark blue) models. Shown is the performance on held-out test data (chromosomes 1, 8 and 21) for 10 biosamples that were not used during pre-training. **(c)** Training curves, showing the area under the precision-recall curve on the validation data (chromosome 9) as a function of the training epoch. The dashed vertical line denotes the training epoch at which the model training is completed. Pre-trained models require fewer training epochs than randomly initialized models and they achieve more accurate predictions.

Predicting the Molecular Effects of Genetic Variants using Interpretation Plugins

- Perform variant annotation and in-silico mutagenesis by contrasting model predictions for the reference allele and for the alternative allele.
- If the model can be applied across the entire genome, such as chromatin accessibility models, sequences centered on the queried variants are extracted.
- If the model can only be applied to regions anchored at specific genomic locations, such as splicing models at intron-exons junctions, only sequences extracted from valid regions that overlap with the variants of interest are used.
- Ease of use for plugins and feature importance algorithms.



```
# Annotate VCF file with
# variant scores
kipoi veff score_variants \
  <Model> \
  --dataloader_args='{
    "fasta_file": "hg38.fa"}' \
  --vcf_path 'input.vcf' \
  -o 'annotated.vcf'
```

```
# Create a mutation map
kipoi veff create_mutation_map \
  <Model> \
  --dataloader_args='{
    "intervals_file": "int.bed",
    "fasta_file": "hg38.fa"}' \
  -o 'mmap.h5'
```

Figure 4: Variant effect prediction and feature importance scores. (a) Schema of variant effect prediction using in-silico mutagenesis. Model predictions calculated for the reference allele and the alternative allele are contrasted and written into an annotated copy of the input variant call format file (VCF). **(b)** Kipoi uniformly supports variant effect prediction for models that can make predictions anywhere in the genome (top) and also for models that can make predictions only on predefined regions such as exon boundaries (bottom). **(c)** Generic command for variant effect prediction. **(d)** Generic command to compute the importance scores using in-silico mutagenesis **(e)** Feature importance scores visualized as a mutation map (heatmap, blue negative effect, red positive effect) for variant rs35703285 and the predicted GATA2 binding difference between alleles for 4 different models. The black boxes in the mutation maps highlight the position and the alternative allele of the respective variant. Additionally, stars highlight variants annotated in the human variant database ClinVar with red: (likely) pathogenic, green: likely benign, grey: uncertain or conflicting significance, other.

Predicting Pathogenic Splice Variants by Combining Models

- Advantages of combining models include: (1) combined scores can cover multiple biological processes, and (2) they are more robust because they average out conflicting predictions of individual models.
- Combined models were i,ii) 5’ and 3’ MaxEntScan8 , a probabilistic model scoring donor and acceptor site regions that was trained on splice sites with cDNA support, iii) HAL9 , a k-mer based linear regression model scoring donor sites that was trained on a massively parallel reporter assay in which hundreds of thousands of random sequences probed the donor site sequence space9 , and iv) Labranchor, a deep-learning model scoring the region upstream of the acceptor site for possible branchpoint locations that was trained from experimentally mapped branchpoints.

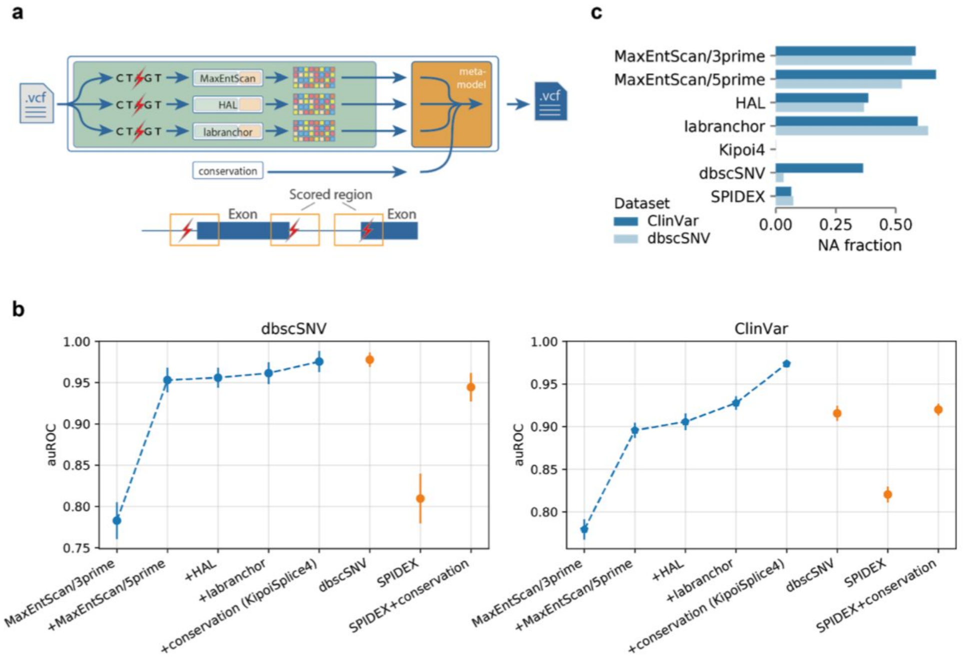


Figure 5: Composite models using Kipoi for improved pathogenic splice variant scoring. (a) Illustration of composite modelling for mRNA splicing. A model trained to distinguish pathogenic from benign splicing region variants is easily constructed by combining Kipoi models for complementary aspects of splicing regulation (MaxEntScan 3' models acceptor site, MaxEntScan 5' and HAL model donor sites, labranchor models the branchpoint) and phylogenetic conservation. These variant scores are combined by logistic regression to predict the variant pathogenicity (orange box). **(b)** Different versions of the ensemble model were trained and evaluated in 10-fold cross-validation for the dbscSNV and ClinVar datasets (Methods). The four leftmost models are incrementally added to the composite model in chronological order of their publication: the leftmost point only uses information from the MaxEntScan/3prime model, while `+conservation (KipoiSplice4)` uses all four models and phylogenetic conservation. These performances were compared to a logistic regression model using state-of-the-art splicing variant effect predictors (SPIDEX, SPIDEX+conservation, dbscSNV). KipoiSplice4 achieves state-of-the-art performance on the dbscSNV dataset and outperforms alternative models on ClinVar which contains a broader range of variants **(c)** Fraction of unscored variants for different models in the dbscSNV and ClinVar datasets.

Outline

Introduction

Approach

Experiments

Discussion

Future Work

References

Discussion

- Unified interface to models, automated installation, and nightly tests.
- Repository and programmatic standard for sharing and reuse of trained models in genomics.
- Pre-computed predictions cannot be extended for new or different input data
- Trained models can be generative, data-modelling distributions. This saves space and time in computing and storing relevant results.
- API contribution brings balance between structure and no structure.

Outline

Introduction

Approach

Experiments

Discussion

Future Work

References

Future Work

- Open challenges for key predictive tasks in genomics with platforms like DREAM or CAGI and make the best models available in Kipoi.
- Continuously update state-of-the-art models.
- More exploration of composite models to capture how genetic variation propagates through successive biological processes.

References

http://kipoi.org/

https://www.biorxiv.org/content/early/2018/07/24/375345

https://github.com/kipoi/models