Phenotype prediction from single-cell RNA-seq data using attention-based neural networks

https://academic.oup.com/bioinformatics/article/40/2/btae067/7613064

ARTIFICIAL INTELLIGENCE AND DATA ANALYTICS LAB
명지대학교 융합소프트웨어학부 김민

Introduction

- Accurate **prediction of phenotypes** is critical in advancing diagnosis, prognosis, and therapy.
- Bulk tissue samples gene expression profiles are measured **by averaging cells across the whole tissue**, which often do **not** reveal the full complexity of **diverse cell types** within patients.

ScRAT (Single-cell RNA-seq Attention-based Transformer)

1. Sample mixup

$$\tilde{x} = \lambda x_i + (1-\lambda) x'_i$$
$$\tilde{y} = \lambda y + (1-\lambda) y'$$

* given two scRNA-seq samples S and S’ * $\lambda \in [0, 1]$

* x_i and x'_i are gene expression profile of cells drawn from S and S’

* y and y’ are corresponding one-hot phenotype label encodings

```
x_mix = lam * x[:batch_size] + (1 - lam) * x_p[:batch_size]

labels_augmented = np.concatenate(
                [labels_augmented, [lam * labels_augmented[idx_1[0]]
              + (1 - lam) * labels_augmented[idx_2[0]]] * diff])
```

ScRAT (Single-cell RNA-seq Attention-based Transformer)

2. Attention layer : 입력 데이터 내에서 서로 얼마나 중요한 관계를 가지는지 계산

- 각 세포의 임베딩 c_i에 대해 변환

$$Q_i = W_q c_i, \qquad K_j = W_k c_j, \qquad V_j = W_v c_j$$

* $W_q c_i$: 현재 기준이 되는 세포 c_i의 Query 벡터 (질문 역할)
* $W_k c_j$: 다른 세포 c_j의 Key 벡터 (대답할 정보)
* $W_v c_j$: 다른 세포 c_j의 Value 벡터 (최종 가져올 정보)

- Self-Attention : 입력 C_i가 다른 C_j에게 얼마나 중요한지를 계산

$$s_{ij} = \frac{dot\ product(Q_i, K_j)}{\sqrt{d_{kqv}}}, \qquad a_{ij} = softmax_j(s_{ij})$$

* d_{kqv} is the dimensionality of key, query and value
* 내적 결과를 **벡터 차원의 제곱근 $\sqrt{d_{kqv}}$ 으로 나눠서 정규화**
* a_{ij} : Attention 가중치

ScRAT (Single-cell RNA-seq Attention-based Transformer)

2. Attention layer

- 새로운 임베딩 생성 (Weighted Sum)

$$h_i = \sum_{j=1}^{N} a_{ij} V_j$$

- Multi-Head Attention : Self-Attention을 K개의 독립적인 공간에서 동시에 수행

$$Attention(c_i) = Concat(h_i^1, \ldots, h_i^K)\, W_0$$

* K different groups

* W_0 : 최종 출력을 조합하는 가중치 행렬

ScRAT (Single-cell RNA-seq Attention-based Transformer)

2. Attention layer

- Randomly select **NC cells** as one fixed-size sample
- Generate **NS fixed-size samples** for each sample

- During the **training** process, each fixed-size sample (NC) is calculated a loss.
- Majority vote to assign the **predicted label** to each sample.

ScRAT (Single-cell RNA-seq Attention-based Transformer)

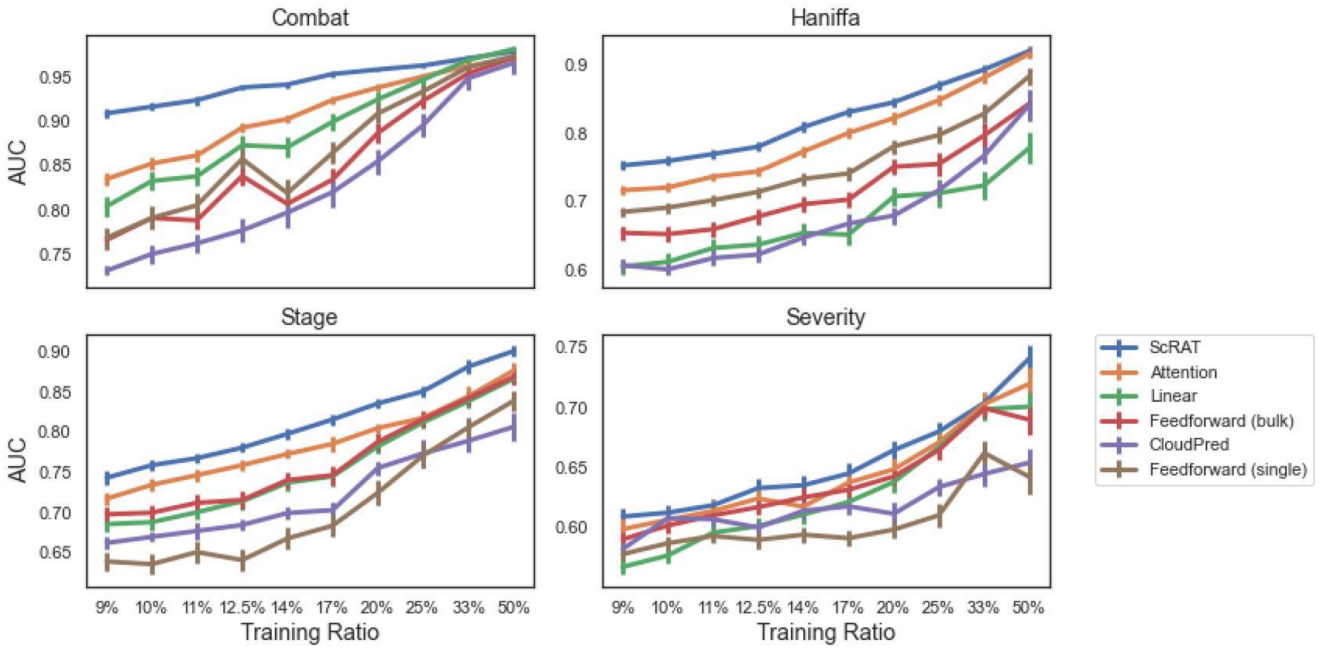
3. Phenotype classifier

- The **cell embeddings** for each sample by computing the **average value** along each dimension.
- The aggregated embedding is passed to the phenotype classifier, **a one-layer MLP**, which outputs the predicted phenotype for the input sample.

Experiment

- For **COMBAT and Haniffa datasets**, we perform the task of disease diagnosis
  - COVID versus Non-COVID
- For **SC4** which includes mostly COVID samples
  - mild/moderate versus severe/critical (경증/중등증 vs. 중증/위중증)
  - convalescence versus progression (회복 vs. 진행)

Results




```
(scrat) kim89@ailab-System-Product-Name:~/ScRAT$ bash run.sh
Namespace(seed=100, batch_size=256, learning_rate=0.01, weight_decay=0.0001, epochs=100, task='stage', emb_dim=128,
 h_dim=128, dropout=0.3, layers=1, heads=8, train_sample_cells=500, test_sample_cells=500, train_num_sample=20, tes
t_num_sample=50, model='Transformer', dataset=None, inter_only=True, same_pheno=-1, augment_num=300, alpha=0.5, rep
eat=1, all=0, min_size=10000, n_splits=2, pca=True, mix_type=1, norm_first=False, warmup=False, top_k=1)
/home/kim89/ScRAT/main.py:317: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a fut
ure version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value
by position, use `ser.iloc[pos]`
  train_ids.append(patient_id[p_idx[i][0]])
======= sample mixup ... ============
100%|| 300/300 [00:10<00:00, 28.28it/s]
cuda
Epoch 1, Train Loss 0.571833, Valid_loss 0.530167
Epoch 2, Train Loss 0.449786, Valid_loss 0.544938
Epoch 3, Train Loss 0.429900, Valid_loss 0.553128
Epoch 4, Train Loss 0.426761, Valid_loss 0.537434
Epoch 5, Train Loss 0.420368, Valid_loss 0.533833
Epoch 6, Train Loss 0.420763, Valid_loss 0.447389
Epoch 7, Train Loss 0.420172, Valid_loss 0.462088
Epoch 8, Train Loss 0.418298, Valid_loss 0.500830
Epoch 9, Train Loss 0.418716, Valid_loss 0.541476
Epoch 10, Train Loss 0.419321, Valid_loss 0.551993
Epoch 11, Train Loss 0.415724, Valid_loss 0.569280
Epoch 12, Train Loss 0.418716, Valid_loss 0.590767
Epoch 13, Train Loss 0.418113, Valid_loss 0.516893
Epoch 14, Train Loss 0.420671, Valid_loss 0.574628
Epoch 15, Train Loss 0.417082, Valid_loss 0.521970
Epoch 16, Train Loss 0.420968, Valid_loss 0.527980
Epoch 17, Train Loss 0.414995, Valid_loss 0.550276
Epoch 18, Train Loss 0.415600, Valid_loss 0.587127
Epoch 19, Train Loss 0.416977, Valid_loss 0.518725
Epoch 20, Train Loss 0.416525, Valid_loss 0.560900
Epoch 21, Train Loss 0.414821, Valid_loss 0.513584
```

```
==================================
=== Final Evaluation (average across all splits) ===
==================================
Best performance: Test ACC 0.840909,   Test AUC 0.903884,   Test Recall 0.821678,   Test Precision 0.796784
```

Conclusion

- ScRAT is designed to learn from limited samples without prior knowledge and provides **accurate phenotype predictions**.
- ScRAT has the potential that suggest **novel molecular mechanisms** and/or **targeted therapies**.