|  |
| --- |
|  |
| **학사학위논문** |
|  |
| **Bulk RNA-seq data를 이용한**  **Immune cell proportion 예측 Deep-learning Model 설계** |
|  |
| **이 건** |
|  |
| **한양대학교 생명과학과** |
|  |
| **2020년 2월** |
| |  | | --- | |  | | **학사학위논문** | |  | | **Bulk RNA-seq data를 이용한**  **Immune cell proportion 예측 deep-learning model 설계**  **지도교수 남 진 우** | |  | | **이 논문을 생명과학 학사학위논문으로**  **제출합니다.** | |  | | **2019년 12월 4일** | |  | | **한 양 대 학 교 자연과학대학** | |  | | **생명과학과** | |  | | **이 건** | |  | |
| |  | | --- | |  | | **이 논문을 생명과학과의 학사학위 논문으로 인준함** | |  | | **2020년 2월** | |  | | **심사위원장 (인)** | |  | | **한양대학교 생명과학과** | |  | |

**Abstract**

tumor microenvironment에 존재하는 tumor infiltrating cell과 tumor의 type이 연관성이 있다는 연구 결과가 존재한다. 따라서 TME 내에 존재하는 cell의 distribution을 파악하는 것은 cancer의 prognosis와 therapy를 결정하는 데 있어 중요한 정보를 제시할 수 있다. 최근 single cell RNA sequencing 기법 등의 개발로 단일 세포의 transcriptome을 분석할 수 있게 되었지만, 한 번에 파악할 수 있는 cell의 개수와 비용 등의 문제로 한계점이 존재한다. 이와 관련하여 본 연구에서는, 기존의 데이터베이스에 존재하는 RNA-seq data과 deep learning 기법을 이용하여 cell fraction을 예측하는 model을 개발하고자 한다.

**Introduction**

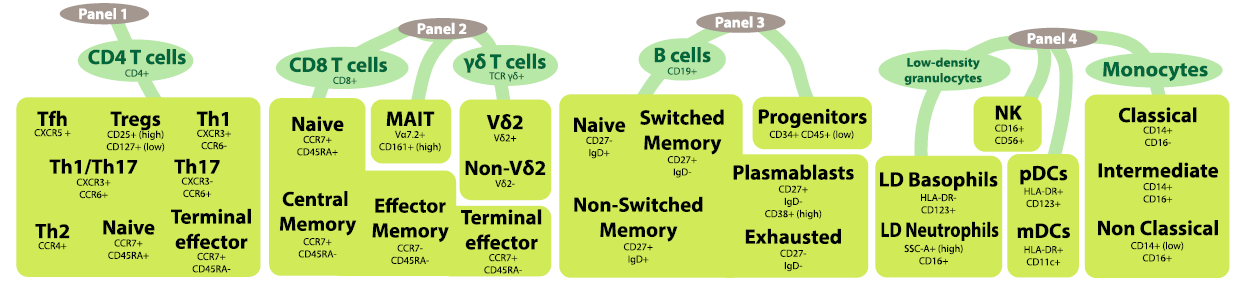
**Tumor의 주변부에는 blood vessel, immune cell, extracellular matrix 등 다양한 complex들이 존재한다. 이러한 tumor를 감싸고 있는 주변 환경을 TME(Tumor Microenvironment) 라고 한다. 최근 논문에 따르면, TME에 존재하는 cell의 type과 distribution는 random하게 있는 것이 아니라, tumor의 type에 따라 특정한 distribution를 갖는다는 것이 밝혀졌다[1][2].** 이러한 cell들 중, tumor infiltrating immune cell들이 tumor control에 중요한 역할을 한다는 것이 밝혀져 있다. 예를 들면, Cytotoxic CD8 T cell은 anticancer immunity에 주요한 effector로 작용한다. 또한 regulatory T cell 같은 경우에는 면역 억제 기능을 해서 tumorigenesis와 immune evasion을 support한다. 따라서, **따라서 TME에 있는 다양한 cell들의 type과 distribution를 파악하는 것은 cancer의 prognosis와 therapy를 결정하는 데 있어 중요한 정보를 제시할 수 있다.**

**최근 Next generation sequencing에 기반한 여러 염기서열 분석 방법들이 개발되면서, tumor의 gene expression을 transcriptome level에서 측정할 수 있게 되었지만, sample 자체에 다양한 complex들이 존재하고 있기 때문에, gene expression이 ensemble로 관찰된다는 단점이 있다. Ensemble gene expression을 나타내는 bulk RNA-seq data를 immune cell의 proportion으로 나타내기 위해서는 deconvolution을 진행해야 한다. 여기서 말하는 deconvolution은** bulk RNA-seq data, 즉 heterogeneous한 sample의 gene expression profile을 특정 cell들의 gene expression level의 convolution으로 간주하고, cell type specific한 expression profile을 갖고 있는 signature matrix를 이용하여 cell fraction을 추정하는 방법을 의미한다.

**이러한 deconvolution을 이용하는 방법 중에, support vector machine을 이용하는 CIBERSORT라는 tool이 존재한다. CIBERSORT에서,** deep learning model의 training에 필요한 데이터는 proportion per immune cell, signature matrix, total gene expression으로 총 3가지이다[3]. 이를 참고하여, 실험에서는 gene expression data를 이용해서 proportion을 예측을 하는 model을 설계해 볼 것이다.

Immune cell proportion은 Monaco 논문의 flow cytometry data에서 얻을 수 있고[4], total gene expression은 논문에 나와 있는 immune cell RNA-seq data와 flow cytometry data를 이용해서, 후에 processing 과정에 있어 immune cell 각각의 RNA-seq data에서 random하게 read를 추출 후 통합하여 bulk RNA seq data를 artificial하게 생성할 수 있다. 이렇게 bulk RNA-seq data를 생성하는 것은, model이 linear하게 training을 진행하는 것을 막아, 추후 deep learning model에서 학습을 깊게 시킬 수 있다고 기대된다. **이러한 gene expression 정보를 이용해서 tumor infiltrating immune cell을 측정할 수 있다면, tumor의 escape mechanism이나 특정 therapy를 적용했을 때의 반응 등을 알아낼 수 있다. 따라서 이러한 정보를 이용할 수 있다면, significant한 clinical effect를 낼 수 있을 것이다.**

**Methods**

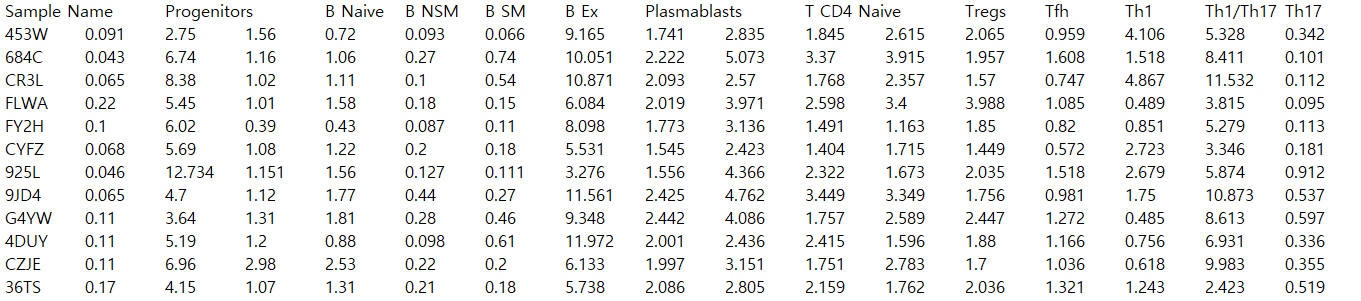


**Figure 1. Representation of Data Collection** (Monaco et al. *Cell reports* 2019)

2 cohorts로부터 나온 PBMC aliquots는, 29개의 immune cell type의 RNA-seq와 microarray data를 생성하는 데 사용되었다.

Deconvolution model을 구현하기 전에 앞서, 딥 러닝에 필요한 데이터를 선정해야 한다. Monaco 논문에서, cell 별로 sorting한 public RNA seq data의 accession number와, 12 patients의 cell fraction이 기입되어 있는 flow cytometry proportion data가 존재한다(figure 1).

**In silico data synthesis**



**Figure 2. Monaco flow cytometry cell fraction data의 일부** 그림에서, 가장 왼쪽 column은 patients의 symbol을 나타내고, 각 column은 각 patients에서 존재하는 immune cell의 proportion을 의미한다.

Deep learning model을 training하는 데 필요한 data의 양은 매우 많기 때문에, 현재 논문에서 얻을 수 있는 immune cell RNA-seq data의 양으로는 model을 training시키기에 부족하다. 따라서 먼저 12명의 patient의 flow cytometry immune cell proportion data에서(figure 2), 각 type의 mean과 standard deviation을 고려하여 100개의 in silico sample을 생성했다. 이 100개의 sample에서, 각 sample의 sum of fraction 값은 90%-97% 사이의 값을 나타내도록 고려하여 진행했다.

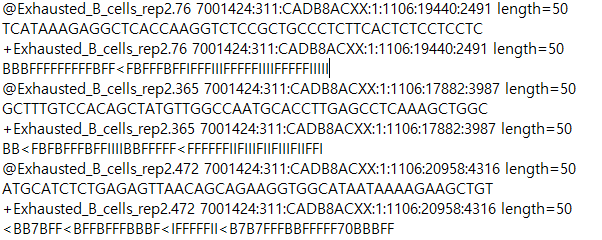
Immune cell 자체가 independent한 상태에서 derived되는 것이 아니라, progenitor cell로부터 differentiation하면서 여러 cell type으로 나뉘는, 이러한 cell lineage를 가지기 때문에, 세분화된 cell type끼리는 굉장히 비슷한 expression profile을 가진다. 따라서 이러한 expression colinearity를 갖는 일부 cell을 merge함으로써, differentiated expression profile을 확보할 수 있다

생성해 낸 100 sample의 29 type immune cell fraction data에서, 이러한 colinearity와 cell fraction을 고려하여, plasmablasts와 switched memory B cell, central memory CD8 T cell과 naïve CD8 T cell, V delta 2 gamma delta T cell과 non-V delta 2 gamma delta T cell, exhausted memory CD8 T cell과 terminal effector CD8 T cell, T helper type 1 cell과 T helper type 17 cell과 T helper type 1/17 cell을 각각 합쳐 총 21개의 variance를 갖는 immune cell fraction data를 생성했다.

**RNA-seq data synthesis**

Monaco 논문에서 제공하고 있는 GEO accession number를 이용해 29 immune cell type과 peripheral blood mononuclear cell의 paired end RNA-seq profile을 이용할 수 있다. 위에서 생성한 100개의 29 immune cell type fraction data를 이용해, 각각의 sample에서 각각의 immune cell이 차지하는 비율을 토대로, Monaco 논문에서 접근할 수 있는 immune cell에 대한 RNA-seq data를 이용하여 새로운 bulk RNA-seq data를 생성할 수 있다.

Monaco 논문의 GEO accession number를 이용해 접근할 수 있는 SRA database에 나와 있는 immune cell RNA-seq data는 terminal effector CD4 T cell을 제외하고, replicate 4개만 제공하고 있다. 때문에 immune cell fraction data를 이용하여 각 replicate 당 25개씩 synthesis를 진행했고(terminal effector CD4 T cell은 replicate 3, replicate 4에서 replicate 1, replicate 2를 사용하였다.), 따라서 총 100개의 새로운 bulk RNA-seq data를 생성했다.



**Figure 3. fastq format의 텍스트 구조.** 1st line은 sequence identifier, 2nd line은 sequence letter, 3rd line은 구분자, 4th line은 sequence에 대한 quality 값을 나타낸다. 이 4가지 line이 반복되는 구조로 파일이 구성되어 있다.

위에서 사용할 RNA-seq data는 fastq format으로 되어 있다. Fastq format에서, 단일 read의 정보는 file 내에서 4 line에 걸쳐 있으므로(figure 3), 이 4 line을 묶고 random sampling하는 작업을 python script로 구현하여 진행했다.

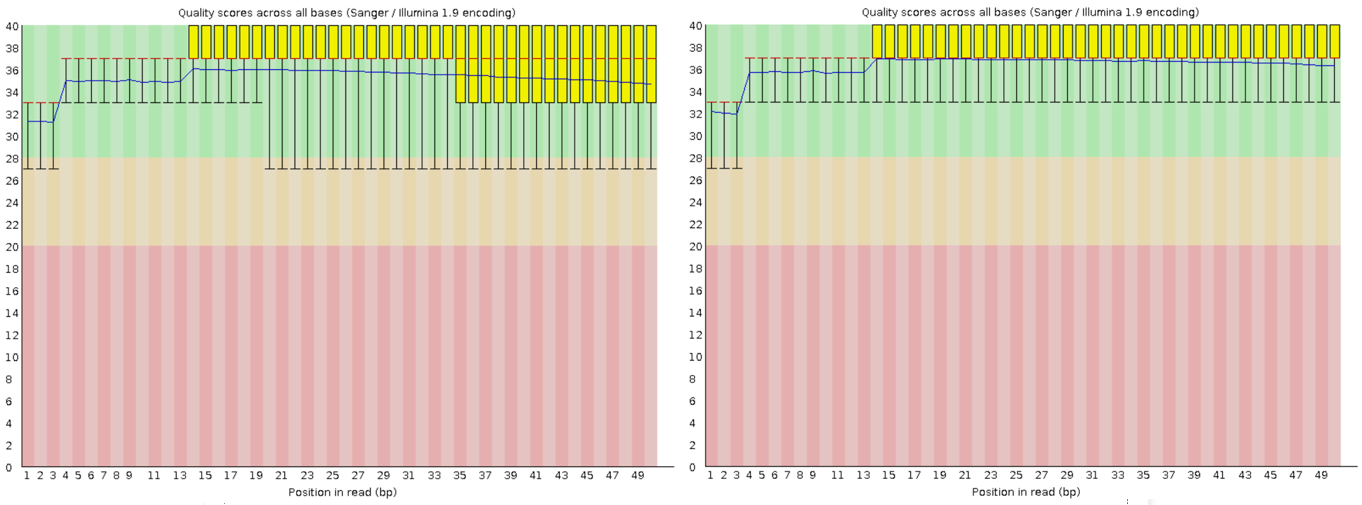
SRA database에 존재하는, GEO107011에 존재하는 immune cell RNA-seq data들은 각각 read 20 million 개의 정보가 담겨 있으므로, 각각의 subfile들은 위에서 정한 proportion에 해당하는 read 개수의 정보만 담겨 있다. 이렇게 random sampling한 각각의 file을 합치는 과정을 통해 하나의 bulk RNA seq data를 만들 수 있고, 총 100 개의 bulk RNA seq data를 생성했다.

**RNA-seq analysis standard operating procedure**

RNA-seq data를 분석하기 위해서는, quantification하는 작업이 필요하다. Gene level에서의 RNA Quantification은 quality control, mapping 과정을 통해 진행할 수 있다.

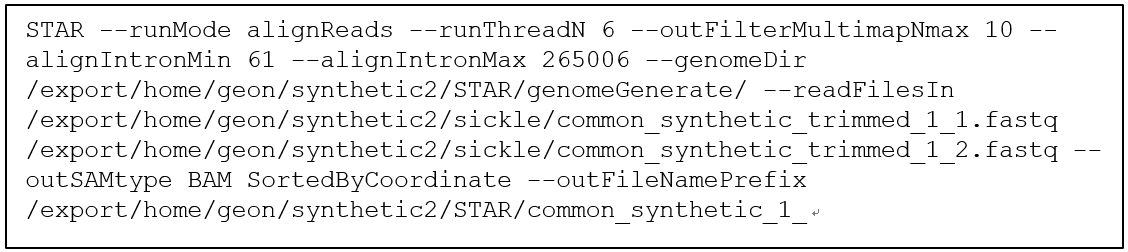
Mapping된 read의 개수만으로 RNA expression quantity를 정의하기에는 sample 별로 sequencing data size가 다를 수도 있고, gene이나 transcript의 length에 따라 mapping된 read의 수도 다르다. 따라서 quantification을 진행하여 나온 read count 결과값을 비교하기 위해서는, 하나의 sample 내의 각각의 값에 대해 normalization을 해 주어야 한다.

일반적으로 bioinformatics에서 사용하는 measurements는 RPKM(Reads Per Kilobase of transcript per Million mapped reads), FPKM(Reads Per Kilobase of transcript per Million mapped reads), TPM(Transcripts Per Million) 등이 있으며, 이러한 measurements 중 TPM이 RPKM에 내재하는 statistical bias를 조정해준다는 장점이 있어 RNA expression quantity를 비교하는 데 더 optimal하다[5].



**Figure 4. FastQC의 결과를 나타내는 그래프** 위의 그래프는 53번째 sample의 read length / quality score 로 나타낸 box plot graph이다. 왼쪽은 quality 보정 전의 read의 distribution을 나타내고, 오른쪽은 quality 보정 후의 read의 distribution을 나타낸다. Box plot이 y축 양의 방향으로 이동한 것으로 볼 때, trimming이 진행되었다는 것을 알 수 있다.

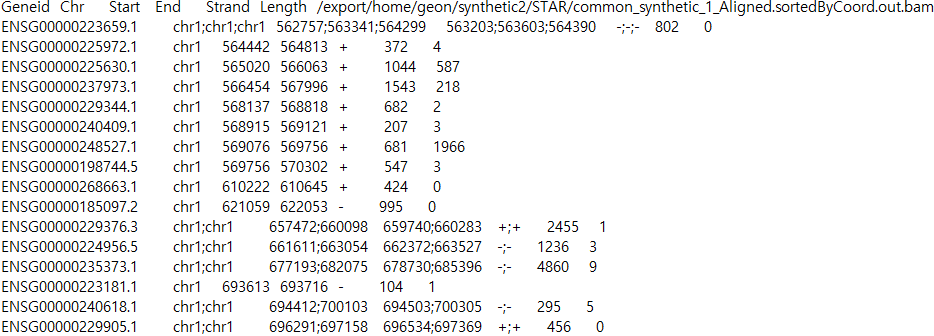
Quality control 단계에서는 read quality를 보정해주는 작업과 확인해주는 작업이 있다. Quality 확인 단계에서는 FastQC라는 프로그램을 사용했으며[6], read의 quality, contamination 정도, length distribution 등의 정보를 시각화하여 확인할 수 있다(figure 4). Quality 보정 단계에서는 sickle이라는 프로그램을 사용했다[7]. Base quality score가 20 미만이고, minimum read length가 20 미만인 read를 cut하도록 설정해서 trimming을 진행했다.



**Figure 5. STAR의 alignReads mode에서의 명령어** GenomeGenerate mode에서 생성한 index file과 fastq file을 이용해서, BAM format의 file을 형성할 수 있다. 맵핑 허용 최소 intron length와 최대 intron length 값은 알려진 human gene의 intron length distribution으로부터 0.01% 구간, 99.9% 구간 값을 설정했다.

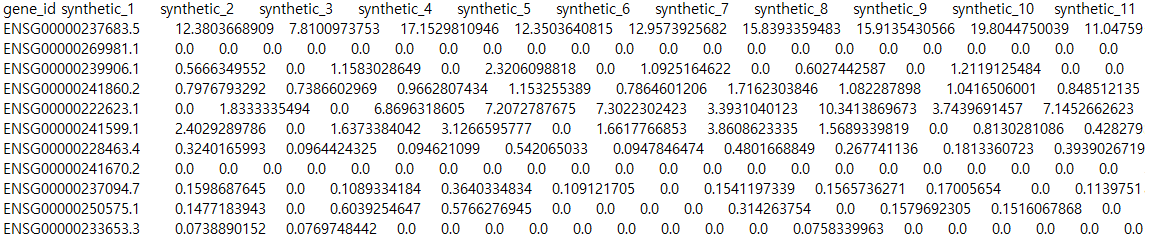
다음 단계로, trimming한 sample들을 STAR를 이용해 참조 유전체에 mapping하는 작업을 진행하였다[8]. STAR에는 2가지 runmode가 있으며, 참조 유전체의 index를 생성하는 genomegenerate mode와 sequencing data를 mapping하는 alignreads mode가 존재한다(figure 5).

Genomegenerate mode에서, index file을 저장할 디렉토리를 설정하고, 참조 유전체 서열이 담겨 있는 데이터와 gene annotation에 관한 정보가 담겨 있는 GTF file을 넣어주어야 한다. 참조 유전체 서열 데이터로는 GRCh37 버전을 사용했고, gene annotation은 gencode v19를 사용했다.



**Figure 6. FeatureCounts를 진행한 결과물의 일부** FeatureCounts는 STAR mapping result를 토대로, 각 gene에 mapping된 read의 개수를 count하는 program이다. 그 결과 gene id, chromosome number, strand information, read count 값 등을 병렬 표기한 파일을 출력한다.

이후 mapping된 데이터는 bam format으로 저장된다. Bam file을 토대로, 하나의 gene 내에 몇 개의 read가 mapped되었는 지 quantification하는 작업이 필요하다. 이는 FeatureCounts라는 프로그램으로 진행했다[9]. 위에서 사용한 gencode v19와 bam file을 input으로 사용했다. 각 gene에 할당된 mapped read의 정보를 담은 파일(figure 6) 외에 이러한 정보들의 summary가 담겨 있는 결과 파일도 생성된다.



**Figure 7. Gene 별 TPM 값으로 나타낸 파일의 내부 모습** 100개 각각의 sample마다 총 57,820 개의 gene에 대한 TPM value를 계산했다.

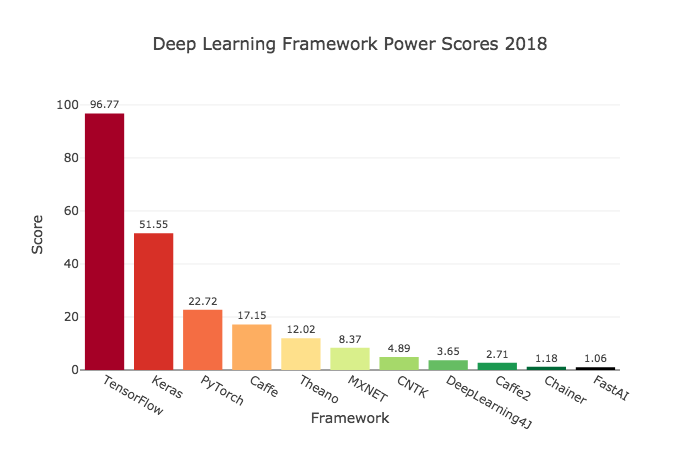
이 quantification result를 갖고 TPM을 계산해야 한다. 그러기 위해, 결과물에서 gene id, gene length와 count를 추출하고, 100개의 sample의 TPM 값을 계산해서 row는 gene id, column은 sample의 정보가 담긴, tab으로 분할된 텍스트 파일(figure 7)을 생성하는 script를 만들어서 실행했다.

이러한 TPM 값을 그대로 쓰기에는, gene의 수가 많아 model을 training시키는 데 많은 시간이 소요될 수 있다. 또한 발현량이 없거나 매우 작은 gene의 경우도 dump data와 마찬가지이므로, 특정 개수의 gene을 선택하는 작업이 필요하다.

따라서, gene을 선정할 때, 각각의 모든 gene id에서 sample 100개의 분산을 구했고, 그 분산이 큰 순서대로, 상위에 있는 gene id를 200개 추출했다. 추출한 ranked gene id를 토대로, 행이 gene id이고 열이 sample인, TPM 값이 수록되어있는 matrix를 구축했다.

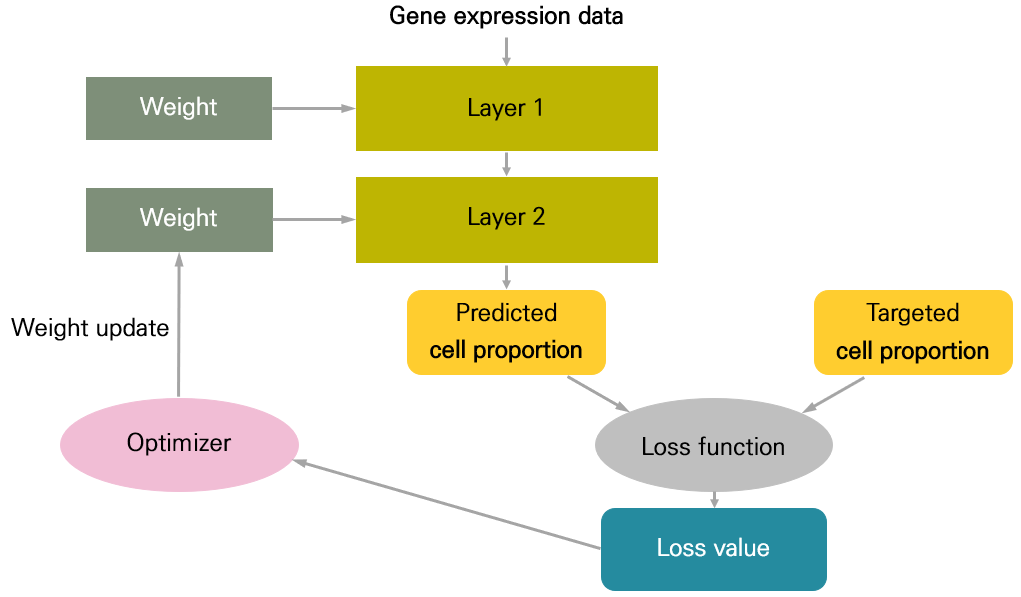
**Deep learning modeling**

Deep learning model을 구현하기에 앞서, 어떠한 framework를 사용할 지를 정해야 한다.

 Room, Chat. "Deep Learning Frameworks." *image* (2018).

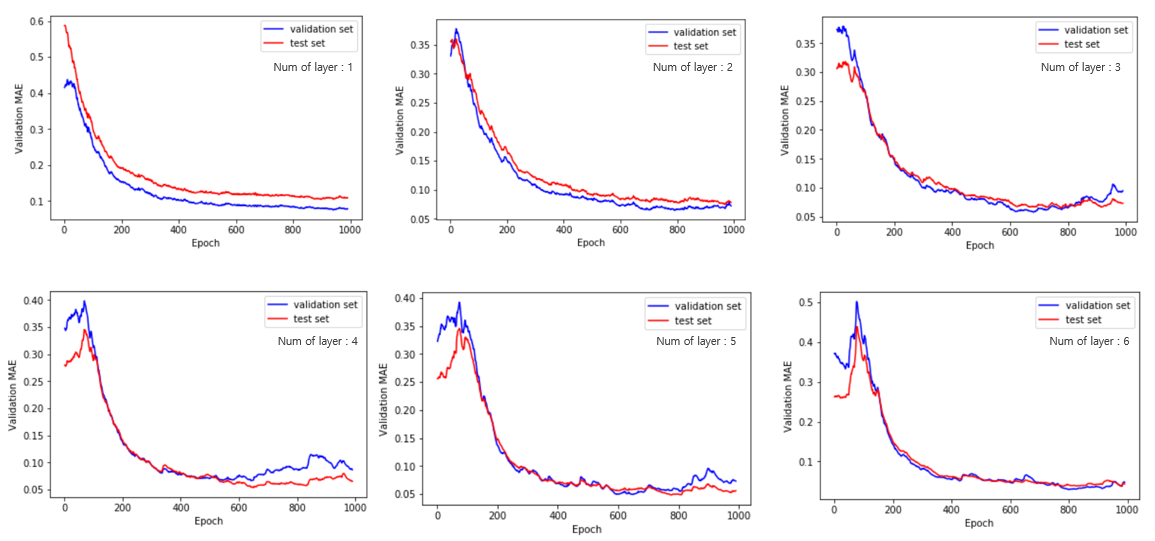
**Figure 8. deep learning framework 사용자 조사** 2018년 KDNuggets 사용자 조사에 따르면, 현재 가장 많이 쓰이고 있는 framework는 Tensorflow이며, 그 뒤를 이어 Keras와 Pytorch의 사용자가 꾸준하게 증가하고 있는 추세를 보이고 있다.

**Top ranked 3 framework를 비교해 보았을 때, 일반적으로 Pytorch가 low level API이기 때문에 big dataset을 다룰 시 가장 좋은 performance를 보인다고 보고되어진다. 하지만 그만큼 architecture가 복잡하기 때문에 전문가 수준의 프로그래밍 실력이 요구된다고 보고된다.** **Keras의 경우**(figure 8)**,** 3개의 framework 사이에서 두드러지게 사용자가 늘어나는 추세를 보이고 있다[10]. **Performance는 pytorch에 비해 낮지만, simple한 architecture를 갖고 있어 사용에 있어 진입 장벽이 낮고, debug가 간편하다는 장점이 있다. 따라서 본 실험에서 model 구현 시 Keras를 사용했다.**



**Figure 9. 구현한 model의 모식도** Immune cell type 별 proportion을 예측하기 위해, 모델은 deep neural network의 가장 기본적인 구현이라고 할 수 있는 multilayer perceptron을 이용해 구현을 할 것이며, 기본적인 구조는 다음과 같다. Activation function은 Relu function, loss function은 mean squared error, optimizer는 stochastic gradient descent로 구현했다. Weight initiation은 Xavier 값을 이용했다.

먼저, 만든 2개의 total gene matrix와 flow cytometry cell proportion data를 이용하여 deep learning model을 설계한다(figure 9). 각 immune cell의 proportion을 predict하는 것이 목표이기 때문에, cell fraction data는 training에서 정답 레이블로 쓰일 것이며, 따라서 input data는 total gene matrix가 될 것이다.

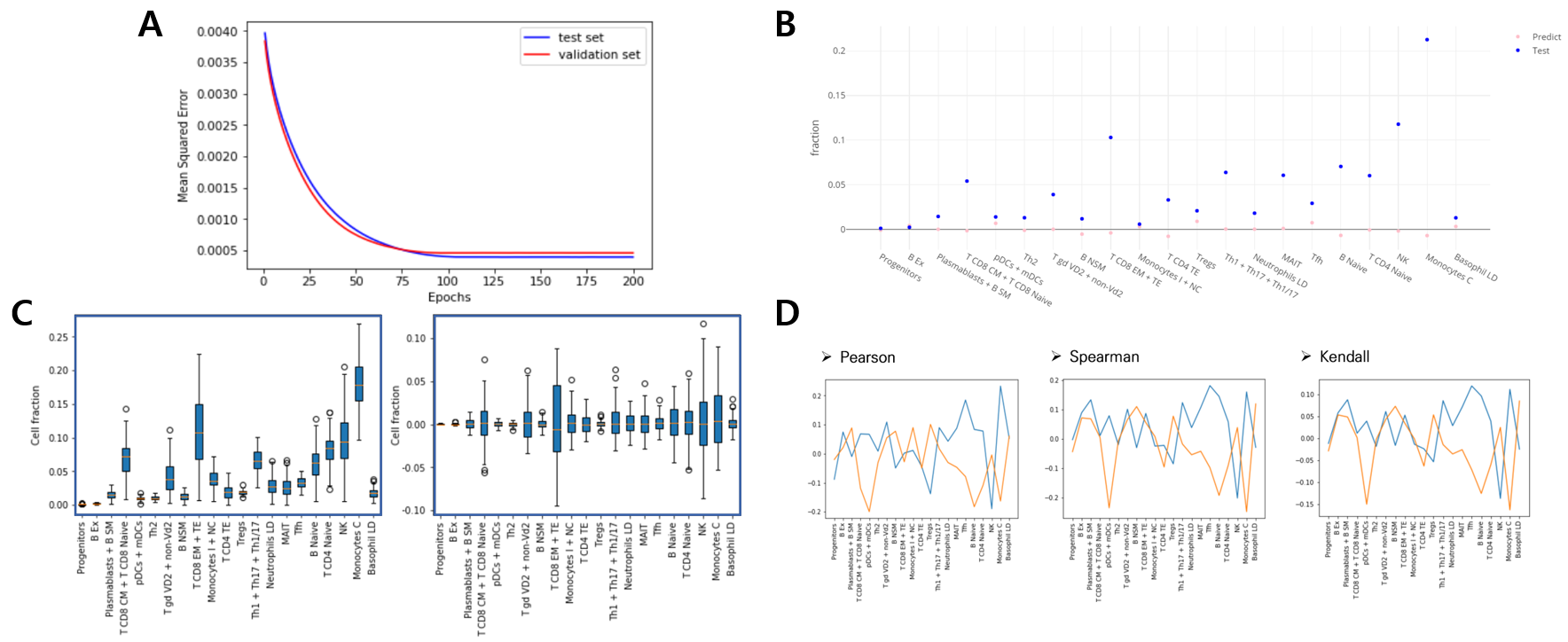


**Figure 10. Hidden layer 수가 다른 model에서 epoch마다의 performance** 각각의 hidden layer에서, 임의로 node 개수를 정해서 validation과 test를 진행했고, 그렇게 만들어진 각각의 model 중 random하게 가져온 결과가 위 사진과 같다. 각각의 정해진 hidden layer 수에서, 30개의 model을 뽑고 1000 epoch동안 학습한 모델의 minimum validation score의 평균을 구한 결과, Hidden layer의 층 수가 2일 때 test data의 performance가 가장 좋은 것을 확인했다.

Network의 층을 결정하기 위해, 제한된 조건에서 실험을 진행했다(figure 10). Input layer는 100개의 node로 구성되어 있고, 각 층의 node는 임의적으로 값을 부여했고, 만든 gene expression data 100 세트 중, 80 세트를 training data로, 20 세트를 validation data로 3-fold cross validation을 진행하였다. Test set은 다른 연구자가 같은 방식으로 만든 데이터를 제공받았다.

최종적으로 설계한 neural network의 전체적인 structure는 각각 1000개, 1개의 node로 구성된 2개의 hidden layer와 21개의 node로 구성된 output layer로 구성된다. 위 모델과 차이점은, L1 kernel regularizer를 각 층마다 구현했고, weight initializer로 normal He 값을 사용했다고, optimizer로 rmsprop을 사용했다는 점이다. Rmsprop은 keras 내의 default 값으로 실행했고, 학습을 70 epoch를 진행했다.

**Results**



**Figure 11. 학습 모델의 정답 레이블 예측 결과**

(A) MSE를 이용한 model의 performance test. Model의 training 진행 정도를 epoch마다 평가한 그래프이다. (B) Cell type 별 fraction value의 분포. 왼쪽은 100개 sample의 cell type 별 fraction, 오른쪽은 100개 sample의 cell type 별 fraction과, 모델에서 예측한 cell fraction과의 차이를 의미한다. (C) Predicted fraction과 targeted fraction을 나타낸 그래프. random하게 추출한 test data를 input에 넣고 예측한 cell fraction과 target cell fraction 값을 비교한 양상 (D) Test data의 Predicted fraction과 targeted fraction의 상관계수. Test data의 Predicted fraction과 targeted fraction의 상관계수 값. Predicted cell fraction과 targeted cell fraction의 correlation 값을 의미한다.

모델을 training 시키는 동안 performance를 보면, 약 70 epoch부터 overfitting이 일어나는 것을 확인할 수 있다(figure 11A). 정답 레이블과 모델에서 예측한 값을 비교해 보았을 때, 값을 제대로 예측하고 있지 못하고 있다는 것을 확인할 수 있었다(figure 11B). 모델의 전체적인 예측 정도를 볼 때, 모든 데이터의 mean을 찾아가게끔 학습이 된다는 것을 확인할 수 있었으나, variance는 큰 차이가 없다는 것을 알 수 있었다(figure 11C). 이는 model이 underfitting되었다는 것을 의미한다고 볼 수 있다. 70 epoch에서 mean squared error로 performance를 측정해본 결과, 값이 5.56\*10-4로 나타났으나 각각의 correlation 값을 보았을 때 predicted cell fraction과 target cell fraction은 상관성이 있다고 판단되지 않는다(figure 11D).

**Discussion**

이러한 결과가 나온 이유를 여러 관점에서 해석해 볼 수 있다. 처음 12명의 patients의 immune cell fraction data에서, 각 cell type마다의 fraction의 mean과 standard deviation을 이용해 random하게 fraction을 생성했는데, 이 fraction이 biological meaning을 가지고 있는 지는 검증되지 않았다.

또한 TPM value에서, sample 간 standard deviation이 가장 높은 gene을 feature gene으로 선정했는데, 이것은 cell type specific gene이 포함되지 않았을 가능성도 존재한다. 또한 noise가 포함되어 값이 크게 나온 TPM 값이 상위 rank에 들어있을 가능성이도 존재한다. 이처럼 cell type specific한 gene이 포함이 되지 않았을 가능성이 있으므로, feature gene을 정확히 뽑아냈는 지에 대한 검증이 되지 않았다.

기본적으로 deep learning model을 training하기 위해서는, data의 개수도 중요하다. 본 실험에서 사용할 수 있는 sample data는 200개로, 일반적으로 deep learning model을 training을 시킬 때의 data보다는 적은 수의 데이터를 사용하고 있다. 또한 model 내의 parameter가 fraction을 정밀하게 predict할 수 있을 만큼 deep하게 구성되어 있는 지는 지금까지 실행한 일련의 과정으로는 알 수 없다.

추후 연구로, cell type 별 gene expression을 나타내는 reference matrix를, NMF analysis와 DEG analysis를 이용하여 feature gene을 토대로 구축하여, neural network와 동시에 학습시키는 방안을 생각해 볼 수 있다. 또한 실험에서 만든 deep neural network를 convolutional neural network로 확장시켜 구현해볼 수 있다. CNN에서의 pooling 과정이 denoising과 비슷한 효과를 볼 수 있도록 하는 과정으로, 데이터들의 주된 feature를 채택할 수 있다고 기대되므로, model의 performance가 높아질 수 있다고 기대된다.

또한 위에서 언급한 TPM denoising 등 좀 더 optimal한 statistical analysis를 통한 data를 대량으로 생성해낼 수 있고, 그를 기반으로 hidden layer를 좀 더 deep하게 구성할 수 있다면, 더 performance가 좋은 model을 구현할 수 있을 것이다.

**Acknowledgement**

실험의 전반적인 진행과 설계, 결과 해석에 있어 조언을 해 주신 한양대학교 생명과학과 남진우 교수님과, RNA-seq analysis SOP를 지도와 겸해 실험 데이터를 제공해 주신 한양대학교 BIG lab 소속 김현우님께 큰 감사를 표합니다.

**Reference**

[1] Junttila, Melissa R., and Frederic J. de Sauvage. "Influence of tumour micro-environment heterogeneity on therapeutic response." Nature 501.7467 (2013): 346-354.

[2] Fridman, Wolf Herman, et al. "The immune contexture in human tumours: impact on clinical outcome." Nature Reviews Cancer 12.4 (2012): 298.

[3] Chen, Binbin, et al. "Profiling tumor infiltrating immune cells with CIBERSORT." Cancer Systems Biology. Humana Press, New York, NY, 2018. 243-259.

[4] Monaco, Gianni, et al. "RNA-Seq signatures normalized by mRNA abundance allow absolute deconvolution of human immune cell types." Cell reports 26.6 (2019): 1627-1640.

[5]Wagner, Günter P., Koryu Kin, and Vincent J. Lynch. "Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples." Theory in biosciences 131.4 (2012): 281-285.

[6] Andrews, Simon. "FastQC: a quality control tool for high throughput sequence data." (2010).

[7] Joshi, N. A., and J. N. Fass. "Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files (Version 1.33)[Software]." (2011).

[8] Dobin, Alexander, et al. "STAR: ultrafast universal RNA-seq aligner." Bioinformatics 29.1 (2013): 15-21.

[9] Liao, Yang, Gordon K. Smyth, and Wei Shi. "featureCounts: an efficient general purpose program for assigning sequence reads to genomic features." Bioinformatics 30.7 (2013): 923-930.

[10] Room, Chat. "Deep Learning Frameworks." image (2018).