Ⅰ. 연구 목표

수많은 질병들과 체내 미생물과의 상관성이 여러 연구를 통해 밝혀지면서 마이크로바이옴 분야가 연구자들의 주목을 받고 있다. 기존 대부분의 연구들은 세계적으로 관심을 가지고 있는 소수 질병에 대해, 정상인과의 차이를 분석하여 인과 관계를 밝혀내거나 명확한 분류를 목적으로 삼는 모델링 연구가 대부분이다. 이러한 연구들은 외부 요인을 명확하게 인식하지 못하고 있을 뿐만 아니라 분석된 데이터 수가 충분하지 않기 때문에 실제 개개인의 질병을 판단하기에 적합하지 않다.
그러므로 본 프로젝트에서는 미생물 커뮤니티 자료를 기반으로 초 고차원 데이터 수집 및 전 치리를 종합적으로 수행한 뒤 어떤 질병까지 동시에 예측할 수 있으며 다양한 환경 조건을 고려한 모델링을 통해 개인 특이적 미생물 분석 방법론까지 이어 나갈 것이다.

Ⅱ. 연구 과정

1. Data collection & pre-processing

Summary

여러 스터디에서 OTU 데이터와 메타 데이터를 수집한 뒤 분석에 용이하게 전 처리를 수행하여 데이터를 준비하는 과정

Detail

미생물 데이터와 그에 해당하는 메타데이터는 미생물 데이터베이스 MGnify와 NCBI에서 수집하였다. 수집한 데이터는 53431 샘플로 총 129개의 서로 다른 프로젝트에서 수집되었으며 57가지의 질병을 포함하고 있다. 연구한 방법론이 적합한지 검증하기 위하여 이

중 일부 데이터만 사용하여 연구를 진행하였다. 수집된 데이터는 분석하려는 과정에 알맞게 아래 순서로 처리되었다.

(1) 각 연구의 OTU 데이터에 phyloseq package를 적용하여 지정된 Taxonomy level로 미생물을 구분시킨다. 추가로 전 처리를 수행한 뒤 모든 연구 별 데이터를 일괄 병합하여mergedOTUdata를 구성한다. 해당 데이터 셋은 taxonomy level별로 병합되었으므로 총 7번의 수행을 거쳐 7개의 데이터 셋을 생성한다. 이 때 사용자의 필요에 따라 특정 Taxonomy 데이터만 생성하여도 무관하다.

(2) 각 연구의 meta data에서 Disease가 없는 샘플 제거, NA 값에 대한 처리, 데이터 타입 통일, 불필요한 컬럼 제거, 불필요한 공백 및 문자열 처리 등 전부 사용자에 의해 직접 전 처리한다. 전 처리된 모든 연구 별 데이터를 일괄 병합하여 mergedMetadata를 구성한다.

(3) 사전에 병합된 두 데이터 셋(mergedOTUdata, mergedMetadata)에서 샘플 별 고유 ID를 기준으로 totalMetaData와 totalOTUData를 재 구성한다. 샘플 중 meta data가 없는 경우, meta data 중 OTU data가 없는 경우는 본 연구에서 분석에 사용할 수 없으므로 제거된다. 마지막으로 모든 샘플에 대해서 0을 가지는 미생물의 경우 분산과 평균이 항상 0이기 때문에 분석에서 불필요하게 차원만 증가시키므로 제거한다.

Code makeMergedOTUdata.R

Code makeMergedMetadata.R

Code makeTotalData.R

2. Find sub stratification

Summary

질병 외 OTU 데이터를 잘 분류시키는 외부 요인(환경 변수)을 찾아내는 과정

Detail

우리가 다룰 데이터는 총 57개의 스터디에서 수집되었으므로 여러 다른 외부 요인(환경 변수)을 포함하고 있다. 하지만 본 연구는 질병에 의한 분류를 목적으로 두고 있기 때문에 OTU 데이터가 질병 외 다른 외부 요인에 의해 분류되는 정도를 최소화시켜야 한다. 해당 단계에서는 OTU 데이터를 너무 잘 분류시키는 외부 요인을 찾기 위해 여러 차원 축소 알고리즘과 클러스터링 알고리즘을 적용하였다.

각 외부 요인 별 OTU 데이터 분류 성능을 확인하기 위해서는 3가지 단계를 거쳐야 한다. (1) 먼저 차원의 저주 이론에 따라 너무 높은 차원에서는 고정된 데이터들의 설명력이 큰 의미를 내포하기 어렵기 때문에 차원 축소 알고리즘 T-SNE, BHT-SNE, PCA, UMAP 네 가지를 사용하였다. 기존 데이터가 가진 분산 정보를 가장 잘 설명할 수 있는 주축을 선택하여 데이터를 축에 투영시키는 PCA 방법은 이상적인 주축의 수를 선택함에 있어 개인 차가 있으며 본 연구처럼 질병을 포함한 여러 외부 요인 각 각이 많은 그룹 수를 가질 경우 클러스터링 성능이 현저히 떨어질 수밖에 없기 때문에 시각화 용으로만 사용한다. 그 다음 T-SNE의 경우 PCA의 단점을 해결해 줄 수 있으나 각 데이터가 이루는 분포에 기반하여 모든 데이터 간 서로의 유사도를 통해 차원 축소를 수행하기 때문에 본 연구처럼 샘플 수가 너무 많은 경우 속도 측면에서 불이익이 생긴다. 게다가 설정하는 parameter인 perplexity에 따라 속도 측면에서 차이가 크다. 그래서 각 데이터에 대한 유사도가 아니라, 거리 기반으로 묶인 그룹 간 유사도로 차원 축소를 수행하는 BHT-SNE를 사용하였다. 하지만 본 연구에서는 차원 축소된 데이터를 사용하는

것이 아니며 단순히 외부 요인 별 구분되는 정도를 목적으로 하고 있기 때문에 차원 축소 간 그룹 간 구조(global structure)와 그룹 내 구조(local structure)를 모두 최소환 잘 보존하는 것이 중요하다 판단하였다. T-SNE는 perplexity에 따라 다르지만 default 기준으로 local structure를 잘 보존하지 못하는 경우가 있기 때문에 최종적으로 두 구조를 모두 일정 수준으로 보존 시켜주는 UMAP을 선택하였다. (차원 축소 알고리즘 선택은 사용자의 판단 하에 결정한다) 이 때 hyper parameter는 default로 두고 수행한다.

(2) 차원 축소를 적용하였다면 이제 축소된 데이터에서 거리 기반의 군집을 나누기 위해 가장 대표적인 클러스터링 알고리즘인 KMeans를 사용한다. 각 외부 요인 별 그룹 수는 대부분 다르며 KMeans는 해당 그룹 수인 K를 지정해주어야 하는 알고리즘 이기 때문에 외부 요인 수만큼 클러스터링을 독립적으로 수행해서 군집 결과를 낸다.

(3) 일반적으로 클러스터링은 target 변수(정답)가 없는 상황에서 데이터 간의 유사도(상관 계수) 혹은 비 유사도(거리)를 기준으로 데이터들을 군집으로 나눈다. 나눠진 군집은 군집 내 데이터 간의 거리, 군집 간의 거리 등을 통해서 silhouette score 혹은 Dunn index 같은 지표 값을 계산해서 군집화 성능을 측정한다. 하지만 본 연구의 경우 이미 외부 요인 별 target 변수 값을 가지고 있으므로 다른 성능 지표를 사용해야 한다. 예를 들어 A, B, C 3개의 그룹을 가진 외부 요인이 있다고 가정해보자. 군집화의 특성 상 아무리 원래 정답 값을 우리가 가지고 있었다 하더라도 군집화 결과에서 어떤 그룹이 A그룹인지 B그룹인지는 연구자의 생각이 개입이 되지 않는 이상 절대 알 수가 없다. 즉 분류 모형처럼 원래 값과 예측 값을 샘플 별로 일 대 일 비교가 불가능하다는 의미이다. 그래서 사용한 지표가 Rand index이다. 원래 같은 그룹 이였던 데이터가 군집화 결과에서 같은 그룹인지 아니면 다른 그룹인지 혹은 원래 다른 그룹 이였던 데이터가 군집화 결과에서도 다른 그룹인지 아니면 같은 그룹인지를 평가하기 때문에 본 연구와

같은 상황에서 적합하다.

하지만 Rand index의 경우 그룹 수가 많아지면 많아질수록 원래 다른 그룹 이였던 데이터가 군집화 결과에서도 다른 그룹이다 라고 판단하는 경우의 수가 압도적으로 많아지므로 항상 성능이 좋게 나올 수밖에 없다. 그러므로 이 문제를 보정하기 위해 나온 Adjust rand index를 최종적으로 성능 지표로 결정하였다.

즉, UMAP + KMeans + ARI를 사용한 것이며 외부 요인 별로 ARI 값을 측정했으므로 이와 같이 외부 요인을 column으로 하는 1개의 행을 가진 ARI table을 만들 수 있다.

Code findSubStratification.R

3. ARI statistical approach

Summary

통계학적 가설 검정을 통해 앞서 찾아낸 외부 요인 결정을 위한 근거 마련

Detail

앞서 UMAP + KMeans + ARI를 통해 1개의 행을 가진 ARI table을 구성하였다. 이제 ARI 값을 통해서 질병 외에 OTU 데이터를 너무 잘 분류시키는 외부 요인을 선택할 수 있게 되었다. 하지만 해당 방법은 측정한 ARI 값이 KMeans의 초기 값의 랜덤화 특성으로 인해 매 수행마다 ARI 값이 변하는 문제를 직면하게 된다. 즉, 1번의 수행만으로 얻은 ARI 값만으로는 어느 외부 요인이 데이터를 잘 군집화 하는가에 대한 답을 낼 수 없다. 이는 모집단과 샘플의 관점에서도 볼 수 있다. 우리는 1개의 샘플을 추출한 것이라 볼 수 있으며, 그 샘플 값만으로는 모집단에서도 각 외부 요인이 해당하는 ARI 값을 가질 것이라고 보장할 수 없다. 물론 비 모수적 가설 검정법에 따라 모집단에 대한 추정이 가능하지

만, 모수적 방법에 비해 통계학적으로 정확한 가설 검정이 불가능하며 우리는 수행을 반복하여 샘플 수를 늘릴 수 있는 상황이므로 굳이 비 모수적 접근을 할 이유가 없다. 즉, 2. Find sub stratification에서 수행한 과정을 여러 번 반복하여 ARI sample 수를 늘려서 Central limit theorem을 만족시킨다면 우리가 수행해야 하는 ANOVA의 기본 조건 중 정규성 검정을 완벽하게 만족시킬 필요가 없다는 근거를 제시할 수 있다. 다시 말해, CLT에 의해 실제 모집단이 정규 분포를 따르지 않더라도 이를 추정하기 위한 통계량인 표본 평균이 근사적으로 정규분포를 따르므로 ANOVA는 정규성으로부터 크게 민감하지 않을 것이고, 이로써 모수적 접근을 위해 정규성을 가정하고 들어가도 된다는 의미이다. 그 외 등 분산성 검정의 경우 상황에 따라 여러 기법이 존재하지만(DDG project manual참고) 본 연구에서는 질병 그룹 간 sample size가 확연히 다르며 정규성이 만족된 것이 아니라 가정한 것이므로 Brown-Forsythe test를 사용했다. 그 결과에 따라 만족하지 못할 시 welch ANOVA를, 만족할 시 ANOVA를 선택하면 되고, 샘플 간의 독립성의 경우 실험자가 아닌 연구자의 입장에서 증명이 불가능하므로 이 역시 가정을 하고 들어간다. 즉, 최소 30번의 재 추출 과정을 반복하여 ARI table을 구성해야 모수적 접근이 허용된다. 물론 이때 많이 반복하면 할수록 이상적이지만 1번의 추출만으로도 외부 요인 수만큼의 2번 과정을 반복하는 꼴이기 때문에 무작정 횟수를 늘리는 것은 적합하지 않다. ANOVA를 수행하면 외부 요인 별 평균 차이의 유무를 파악할 수 있으며 만약 귀무 가설을 기각했을 경우 어떤 집단 간의 차이인지 파악하기 위해 사후 분석을 수행하면 된다. 사후 분석에는 여러 가지 방법이 있지만 본 연구에서 의미가 있는 두 방법만 비교해보았다. Tukey HSD test는 모든 집단 간의 차이에 대한 분석 결과를 보여주는 검정법으로 집단의 표본 수가 동일할 때만 사용 가능하기 때문에 본 연구에 적합해 보인다. 하지만 표본 수가 적을수록 검정의 정확성이 떨어지며 질병과 다른 외부 요인과의 차이만 필요한 상황이므로 sample size와 equal variance에 robust한 Dunnett's test를 사용하였다. 그 결과

질병의 평균 ARI 값과 차이가 거의 없는 외부 요인을 통계학적 근거를 바탕으로 결정 지을 수 있었다.

Code findSubStratification.R

4. GLM modeling

Summary

모델링을 통해 공 변량을 보정함과 동시에 질병 간 유의미한 차이를 보이는 미생물 추출

Detail

앞선 단계에서 질병 외 OTU 데이터 값에 의해 잘 분류되는 외부 요인을 결정지었다. 본 연구의 경우 Bio project와 body product가 이에 해당한다. 이처럼 수집된 수많은 외부 요인들 중 OTU 값을 잘 분류시킨다는 것은 결국 OTU 값 결정에 큰 영향을 미치는 외부 요인이라 할 수 있다. 우리는 서로 다른 여러 연구의 데이터를 합쳤기 때문에 OTU 값은 각 자가 실험된 방식 혹은 환경(외부요인)에 따라 편향되는 값을 가질 수밖에 없다. 예를 들어 A가 침에서 발견되는 대표적인 미생물일 때 침과 소변에서 A라는 미생물을 추출하였다고 가정하자. A는 당연히 침에서 많을 것이고 소변에서는 적을 것이다. 이러한 차이는 분명 질병에 의한 차이가 아니며 미생물 추출 부위에 따른 차이라 말할 수 있다.

즉 OTU를 종속변수, 선택된 외부 요인을 독립 변수라 봤을 때 위 예시처럼 종속 변수와 공유하는 변량을 가진 변수 = 종속 변수에 과하게 영향을 주는 변수 = 다른 독립 변수 자체의 순수한 영향을 측정할 수 없게끔 방해하는 변수를 공 변량이라 부른다. 본 연구의 목적에 따라 OTU 값을 최대한 질병만으로 분류시키기 위해서는 공 변량의 영향력을

반드시 보정해 주어야 한다.

일반적으로 공 변량은 예측 성능을 높여 주기 때문에 모델에 항상 포함시키는 것이 유리하다. 반면 공 변량에 대한 보정이 필요할 때도 모델에 포함시켜야 한다. 방식은 전자와 후자 모두 동일하지만 결론적으로 우리가 필요로 하는 값이 다르다. 전자의 경우 모델로 예측한 값에 대한 정확성을 높이기 위한 공 변량 포함 이였다면 후자의 경우 모델로 설명할 수 없는 정도를 나타내는 Error term에 대한 값을 얻기 위한 공 변량 포함이다. 쉽게 말해 공 변량 보정을 위해서는 공 변량을 포함한 모델을 만든 뒤 residual 값만 가져오면 된다. 이 residual 값을 기반으로 ANOVA를 수행하면 공 변량 보정이 이루어진 값을 바탕으로 특정 그룹 간 비교가 가능해진다. 이것을 ANCOVA라고 부른다. 기존에 우리는 ANOVA, t-test와 같은 개념들이 엄밀히 말하면 Regression에 포함되는 개념인 것을 알고 있다. Categorical 독립 변수와 Continuous 종속 변수를 가진 simple regression에서는 위 두 개념과 유사하게 그룹 간 평균 비교가 가능하다는 것이다. (물론 완전히 동일한 것은 아니다. Regression에서는 baseline 개념도 존재하며 multiple이 되었을 경우 얘기가 완전히 달라진다.) 즉 ANCOVA 역시 Regression approach 관점에서 해석될 수 있다. 자세한 과정은 링크를 참고하고 결론을 말하면, 회귀식에 필요한 종속 변수와 공 변량을 모두 포함시킨 뒤 모델을 수행할 때, 회귀 식의 각 항의 값을 적절하게 조절하면(design matrix) Regression으로도 ANCOVA를 수행할 수 있게 된다. 정확히 말하면 Two-way ANCOVA를 고려해주어야 한다. (반응 변수가 2개 이상일 경우 MANCOVA 고려, ANOVA VS ANCOVA VS MANOVA VS MANCOVA)

About GLM model 이제 공 변량 보정에 앞서 사용할 적절한 회귀 모델을 선택해야 한다. 회귀 모델에는 우리가 기본적으로 사용하던 linear model이 가장 보편적이다. 하지만 linear model은 사실 종속 변수가 정규 분포를 어느 정도 따를 때 정확한 예측이 가능한 모델에 불과하다. 대부분의 실제 데이터에서는 정규 분포를 만족시키는 경우가 매우 드

물기 때문에 다른 분포를 가정하는 Generalized linear model을 사용해야 한다. Generalized linear model은 종속 변수의 타입에 따라 link function을 선택하여 하나의 직선이 아니라 특정 분포를 따르는 곡선으로 모델링을 수행하는 것이다. 데이터 타입이 count value라면 poisson or negative binomial, two group categorical이라면 binomial 등 종속 변수가 따르는 분포와 동일하게 (종속변수=) linear regression의 형태를 변형시켜주는 link function들이 존재한다. 이렇게 데이터 타입을 통해 대략적인 분포의 형태를 가정했다면 이제 가정한 분포가 가진 파라미터를 파악하여 데이터와 확률적 관점에서 조금이라도 더 유사한 분포를 찾아내야 한다. 이 때, 앞서 말한 파라미터는 분포의 형태 혹은 퍼진 정도를 결정짓는 값을 의미하고 다른 말로 "dispersion parameter" 라고 부른다. 본 연구를 예시로 들면 종속 변수 OTU 데이터는 미생물의 수를 의미하는 count value이므로 기본적으로 poisson distribution을 가정하지만 명확히 poisson인지 아니면 poisson과 유사한 Quasi poisson인지 아니면 negative binomial인지 알 수 없다. 그래서 poisson 분포가 가진 파라미터=dispersion=λ를 추정해서 비교적 정확한 분포를 찾아내야 하는 것이다.

먼저, 명확히 poisson이 아니라는 사실은 이미 알고 있다. poisson은 Mean = Variance를 기본 조건으로 삼기 때문에 OTU 데이터 같은 over dispersion(Variance > Mean) 값에는 적합하지 않다. 그래서 우리는 Quasi-Poisson 혹은 Negative binomial 중 한 분포를 선택하기 위해 poisson에 대한 dispersion parameter를 구했고 위 링크 원리에 따라 negative binomial regression으로 결정하였다. Negative binomial dispersion parameter 추정은 대표적으로 Maximum Likelihood Estimator, Method of Moments Estimator를 사용한다. 결국 추정된 parameter의 분포를 기반으로 모델을 선정하였다. 해당 모델에 우리가 가진 데이터를 fit 시키면 이제 모델이 우리의 데이터를 얼마나 잘 설명해 낼 수 있는지 판단하기 위한 성능 평가 단계를 거쳐야 한다. 기존 linear model의 경우 실제 값과 모델 예

측 값의 차이를 통해서 MSE, RMSE, MAE 등으로 residual를 계산했다. Generalized linear model 역시 이와 유사하게 차이 값을 추정 값으로 정규화 하여 계산하는 working residual, pearson residual, response residual이 존재한다. 반면 단순히 추정 값으로 정규화 하여 residual를 구하는 것이 아니라 glm에서 모델이 데이터를 얼마나 잘 fit했는지를 표현하는 deviance로 정규화 하여 residual를 계산하는 deviance residual = residual deviance가 일반적으로 사용된다. 즉 glm에서 deviance는 residual을 구하거나 모델에 대한 적합도 검정을 위한 보편적인 지표이다.

edgeR package를 통해 generalized linear model 중 하나인 negative binomial regression을 수행하는 것은 5가지 과정으로 나뉜다.

(1) Design matrix 구성

쉽게 말해 만들 모델에서 필요한 각 변수들을 직접 디자인하는 것이라 생각하면 편하다. 예를 들어 연속형 반응 변수 y, 연속형 설명 변수 x_1에 대해 다음 회귀 식을 가정하자. $y = \beta_0 + \beta_1 x_1$: x_1은 연속형 이므로 1개의 항으로 표현될 수 있다. 하지만 만약 x_1이 이산형 문자열 설명 변수라 가정해보자. 그러면 1개의 항으로 그룹 별 문자열을 표현할 수 없기 때문에 문자열을 숫자형으로 바꿔주어야 한다. 이 방식을 one-hot encoding이라 한다. 이 때 각 그룹이 ordinal인지 nominal인지에 따라 dummy variable 방식을 사용할지 label variable 방식을 사용할 지 결정한다. 본 연구에서는 질병, 프로젝트 명, 추출 위치 등 대부분의 외부 요인 변수들이 nominal 이므로 dummy variable을 사용했다.

추가로 design matrix를 만들 때는 추후 본인의 연구 방향성에 맞게 intercept를 포함하거나 제거해주어야 한다. Intercept를 포함하게 되면 특정한 baseline이 생기기 때문에 어

떤 정해진 한 그룹과 각 각의 나머지 그룹 간의 비교에는 유용할 수 있으나 그 외에는 오히려 해석이 어려워지는 등의 문제가 생기게 될 수도 있다. 반면 intercept를 제거할 경우 contrast test를 통해 어떠한 비교에도 제한이 생기지 않겠지만, 만약 모델에 이산형 설명 변수가 2개 이상 존재할 경우 2번째 설명 변수 이후로는 변수 별 그룹이 1개씩 누락된다. 예를 들어 y=group(2) + strain(3) 이 있을 때, dummy variable를 통한 design matrix는 다음과 같다.

y=group(2)+strain(3)	group1	group2	strainA	strainB	strainC
s1	1	0	0	1	0
s2	1	0	1	0	0
s3	1	0	0	1	0
s4	0	1	0	0	1
s5	0	1	0	0	1
s6	0	1	0	0	1

위 표처럼 두 이산형 설명 변수의 모든 그룹으로 design matrix를 구성하게 되면 (2) 문제가 발생하게 된다. (물론 이 때만 생기는 문제는 아님) 모든 샘플들은 반드시 항상 group1과 group2 중 1개에 속해 있으며 동시에 strainA, B, C 중 1개에 속해 있다. 그러므로 모든 그룹을 사용하여 design matrix를 만들게 되면 항상 group1 + group2 = strainA + B + C가 되기 때문에 문제가 되며 그 문제는 아래에서 다룬다.

(2) Multi Collinearity 문제 및 interaction term

다중 공선성은 회귀 모델에서 설명 변수 간에 강한 상관 관계에 의해 발생하는 문제로 연속형 변수에서는 잘못된 계수의 추정에 그치지만 이산형 일 경우 glm에 fit조차 되지 않는다. 위 그림을 예로 들어보자. Group1 = strainA + strainB이며, Group2는 strainC 인 것을 알 수 있다. 다시 말해 strainA에 속한 샘플들의 반응 변수 값의 평균과 strainB에 속한 샘플들의 반응 변수 값의 평균의 전체 평균은 결국 group1에 속한 샘플들의 반응 변수 값의 평균과 동일해진다는 것이다. 이렇게 되면 샘플들의 반응 변수 값이(본 연구에서는 OTU 값) group1의 영향력을 받은 값인지 아니면 strainA 와 B

의 영향력에 의한 값인지 전혀 알 수가 없게 된다. 그래서 group1과 strainA, B는 서로 강한 linear dependency를 갖는다고 하며 이 때 영향력을 판단할 수 없는 문제를 다중 공선성이라 정의한다. 이를 다른 말로 "Design matrix not of full rank" 이라 한다. 앞서 (1)에서 언급한 문제도 같은 문제이므로 각 변수의 한 그룹이 누락되는 것은 이와 같은 이유 때문이다. 이 문제를 해결하기 위해서는 이산형에서 하나의 그룹을 누락시키는 방법도 있었지만 그럼에도 다시 다중 공선성이 발생할 경우 해당하는 설명 변수를 모델에서 제외시켜 주어야 한다. 아예 제외시키는 방법 외에 linear dependency를 가지는 두 공 변량에 대해 interaction을 해주는 방법도 있다. Group1에 속하는 샘플은 항상 strainA 혹은 strainB에 속하며 Group2에 속하는 샘플은 항상 strainC에 속하기 때문에 group과 strain을 하나의 변수로써 여기겠다는 의미이다. Interaction은 다중 공선성 문제를 최소화해줄 수 있으며 R에서는 : 로 사용한다. 위 예제에서는 group1strainA, group1strainB... group2strainC로 총 2 x 3 = 6개의 그룹을 가지는 하나의 변수를 사용하게 되는 것이다. 하지만 본 연구에서는 interaction을 고려하게 되면 추후에 수행하게 될 ANOVA F-test와 contrast test에서 질병 간 차이를 보이는 유의미한 미생물 추출에 제한을 주기 때문에 고려하지 않았다. 그러므로 본 연구에서는 의존성을 가지는 공 변량을 제외시키기만 했으며 보정할 수 없는 공 변량으로 결정지었다.

(3) Normalization

Biology 데이터에 대한 정규화 방법에는 여러 가지가 있다. 대표적으로 TSS, CSS (Method부분)를 사용하지만 본 연구에서는 TMM 알고리즘 기반의 logCPM 방법으로 정규화를 하였다. TMM 알고리즘을 이해하기 위해선 Trimmed mean과 Weight mean에 대한 이해가 선행되어야 하며, 구체적인 방법은 TMM 논문의 [TMM normalization details] 부분을 참고한다.

(4) Estimate dispersion

앞서 About GLM model에서 언급한 바와 같이 (1)에서 만든 design matrix를 기반으로 모든 샘플들에 적합한 특정 분포를 파악하기 위해 dispersion을 추정하는 과정을 거친다. 이 때 design matrix를 고려하는 이유는 샘플들이 여러 설명 변수에서 어느 그룹에 속해 있는지 파악하고 구한 평균 값을 기반으로 추정하기 위함이다.

(5) Negative binomial model fit

(4)에서 추정한 dispersion parameter를 갖는 negative binomial distribution을 가정하는 glm model를 생성한다. 이 후 공 변량이 보정된 OTU 데이터가 필요하므로 residual 값을 추출했다. (?)

Hypothesis test on model

준비된 OTU 데이터에 적합한 Generalized linear model을 구축했다면 이제 모델 위에서 다양한 통계학적 가설 검정을 통해 특정 질병 간 차이를 보이는 미생물 추출, 모든 질병 간 비교에서 유의미한 미생물 추출, 특정 질병과 나머지 질병 간 차이를 보이는 미생물 추출 등을 수행할 수 있다. 주의해야 할 점은 design matrix를 만들 때와 마찬가지로 각 항에 들어가야 할 값들을 상황에 맞게 조절해야 하기 때문에 본 연구에서는 intercept를 제거해주었다는 점을 생각하자.

EX) OTU = disease + sex

→ OTU = diseaseA + diseaseB + diseaseC + sexFemale + sexMale

→ OTU= $\beta_1 diseaseA + \beta_2 diseaseB + \beta_3 diseaseC + \beta_4 sexMale + \beta_5 sexFemale$

(1) ANOVA F test

주어진 회귀식에서 coef=1로 두었다면 diseaseA에 해당하는 샘플들의 OTU 평균 값이 계수로 정해진다. (intercept가 있었다면 다르다) 하나의 coef만 지정했다면 귀무 가설은 diseaseA=0으로 가설검정을 수행한다. 하지만 우리가 알고 싶은 것은 모든 disease에 대한 ANOVA F test이다. 이를 수행하기 위해선 귀무 가설이 diseaseA=diseaseB=diseaseC=0이 되어야 하며 sexFemale과 sexMale은 design matrix를 따라야 한다. sexFemale과 sexMale까지 coef를 지정해주게 될 경우 결과는 disease 그룹 간 차이가 아니라 disease와 sex까지 모두 고려한 결과가 도출되게 된다.

다시 말해 두 개 이상의 categorical 설명 변수가 있을 때 특정 변수에 대한 ANOVA F test를 하려면 해당 변수의 그룹 수에 해당하는 항 들에만 coef를 지정해야 하며, 나머지는 design matrix를 따라가게끔 두어야 한다. 그 결과 disease 그룹 간의 variance와 각 disease 내의 variance를 고려한 F 및 p-value 값을 도출하여 모든 disease 간의 유의미한 미생물 추출이 가능하다. 이 때 위 과정은 모든 미생물(column vector) 각 각에 대해 수행되므로 F-value가 높은 미생물 순으로 feature importance를 결정하였고 이를 기준으로 feature selection을 수행하였다.

(2) Contrast test

Contrast test의 경우 가설 검정의 목적에 따라 종류가 다양하다. 본 연구에서는 추후 분류 모형에서 비교적 분류 성능이 떨어지는(sensitivity 기준) 질병에 대해 그 질병에만 유의미한 영향을 주는 미생물을 추출해서 분류 성능을 보정하기 위함 이므로 하나의 질병 VS 모든 질병에 대한 contrast test가 수행되어야 한다.

위 회귀 식의 경우 disease에 3개의 그룹이 존재하므로 contrast를 1, -0.5, -0.5의 조합으로 만들면 된다. (1,-0.5,-0.5) 일 경우라면 diseaseA VS disease B&C 가 되는 셈이다. Disease에 대한 contrast 설정은 마쳤고 이제 남은 sexFemale과 sexMale에 대한 contrast

도 지정을 해주어야 한다. Contrast는 회귀 식에서 각 항에 들어가는 값이 아님을 알아야 한다. 각 항에 들어가는 값은 design matrix 값이며 contrast는 단순히 그룹 별 평균에 대한 가중치를 어떻게 줄 것인가를 의미한다. 즉 평균 비교를 하려는 변수는 disease이므로 sex에는 contrast는 지정하지 않아야 원하는 평균 비교가 가능하므로 sexFemale=0, sexMale=0으로 둔다. 다시 말해 이 예시에서는 sex만 있었지만 disease를 제외한 모든 외부 요인(공 변량)들은 contrast를 0으로 두어야 한다. 만약 어떤 샘플이 disease가 A그룹이며 male이였을 경우 design matrix에 의해 아래와 같은 회귀 식이 만들어진다.

OTUtmp= $\beta_1 1 + \beta_2 0 + \beta_3 0 + \beta_4 1 + \beta_5 0$

이 때 OTUtmp 값은 diseaseA의 샘플로 분류될 것이고 이와 같이 모든 샘플을 거쳤을 때 diseaseA의 OTU 평균을 명시하는 β_1을 구할 수 있으며 이 때 OTUtmp는 β_1을 결정하는 한 요소가 될 것이다. 이렇게 diseaseB와 diseaseC에 대해서 β_2, β_3 까지 값이 결정되면 비로소 A, B, C 세 질병 간 평균 가중치를 적용하여 비교하는 contrast test가 가능해진다. 그러므로 본 연구에서도 특정 disease VS 모든 disease에 대한 contrast test를 수행할 때 이 외의 항들은 모두 0으로 설정한 뒤 진행하였다.

결론적으로 어떤 한 질병에서 유의미한 차이를 보이는 미생물을 찾아내었고, 해당 과정은 모든 질병에서 동일하게 수행하였다.

Code glmModeling.R

5. Classification modeling

Summary

추출된 미생물을 바탕으로 분류 알고리즘 수행 및 특정 질병에 대한 성능 보정 수행

Detail

이제 OTU 데이터로 질병에 대한 classification을 수행한다.

하지만 이에 앞서 본 연구처럼 미생물 수가 너무 많은 초 고차원의 경우 차원의 저주 이론에 따라 정확한 분류를 방해할 수 있으며 시간적인 면에서도 손해가 크다. 그래서 feature selection 과정이 수반되어야 한다. (feature extraction은 feature에 대한 해석이 불가능함으로 적합하지 않다)

이미 앞선 과정에서 ANOVA F-test 기반의 F-value를 통해 feature importance 순으로 나열한 정보가 있다. 해당 정보로 feature selection을 하기 위해서는 F-value가 어느 수준 이하일 경우 쳐낼 것인지 결정해야 한다. 하지만 그 기준은 매우 주관적이고 모호하기 때문에 섣불리 결정할 수 없다. 게다가 일반적으로 feature 수가 많으면 많을수록 분류 성능이 좋을 수밖에 없으며 분류 알고리즘마다 이상적인 feature 수는 다를 수밖에 없으므로 feature importance 기준으로 (전체 미생물 수 – 모델을 위한 최소 미생물 수) / 15를 계산한 값 x의 배수로 feature selection을 진행하였다. 분류 알고리즘 실행 횟수를 15회+-1로 제한하기 위해 위와 같이 계산하였으며, 최소 미생물 수는 임의로 20으로 설정하였다.

즉 1개의 분류 알고리즘에 대해 처음에는 중요도 높은 미생물 20+x개를 선택, 그 다음에는 20+2x개, 20+3x개… 총 미생물 수 순으로 해서 accuracy가 높은 feature의 수를 알고리즘 별로 조사하였다. Accuracy는 높으면서 feature 수는 낮은 것이 이상적이기 때문에 max(accuracy)와 0.05이상 차이가 나지 않는 최소의 미생물 수를 선택하였다.

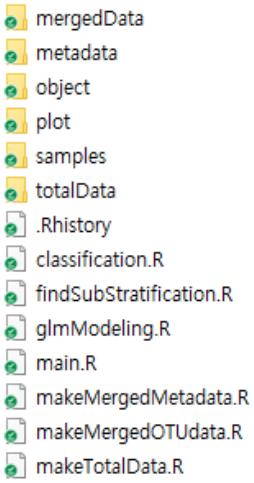
이 때 분류 알고리즘은 3 repeats 10-fold cross validation으로 학습 및 검증 단계를 거쳤다.

최종적으로 알고리즘 별 최적의 feature 수를 기준으로 다시 분류 모델을 수행하여 전체 accuracy와 질병 별 sensitivity를 추출하였다. 선택된 미생물들은 전체 질병 분류에 최적화 되어있기 때문에 특정 질병에만 유의미한 영향을 주는 미생물이 포함되지 않았을 가능성이 있다. 따라서 분류 과정에서 sensitivity가 유난히 낮았던 질병에 대해, 앞서 수행한 contrast test를 기반으로 추출된 미생물을 추가하여 분류 성능을 일부 보정하였다. 보정은 일부 알고리즘에서만 수행되었으며 전반적으로 모든 질병에 대해 sensitivity가 동등하게 높거나 낮은 경우에는 따로 보정을 해주지 않았다.

Code classification.R

Ⅲ. 코드 설명

1. Basic setting



모든 R 파일은 동일한 디렉토리에 위치해야 하며 오른쪽과 같이 6개의 디렉토리를 추가로 생성해 두어야 한다. 각 R 파일에서 path는 본인이 설정해야 한다. [samples] 디렉토리에는 여러 연구에서 사용된 sample들의 OTU 데이터만 포함되어야 하며, [metadata] 디렉토리에는 해당 sample들과 연관된 메타 정보 데이터만 포함되어야 한다. Sample과 metadata는 MGnify와 NCBI에서 수집 한 것으로 따로 제공되지 않는다.

최종적으로 분석에 사용한 데이터는 [totalData]에 있으며, 각 분석 과정의 결과는 함수 리턴 값 자체를 [object]에 저장한다. 그 외 추가적인 시각 자료들은 [plot]에 저장된다. 본 연구의 결과인 [object] 의 R data는 제공되지 않으며 [plot] 의 시각화 자료들은 result절에 첨부하였다. (* [ ]는 directory를 의미한다)

2. makeMergedOTUdata.R

OTU.table

Description

Make OTU data automatically with phyloseq, merge all samples and pre-process the data

Usage

```
OTU.table(dataPath, taxa="Genus")
```

Arguments

dataPath	Your directory path which contains whole sample's OTU data
taxa	Specify taxonomy level which you want to return. Options are "Genus" "Family" "Order" "Class" "Phylum"

Value

Returns the merged OTU data into the designated taxonomy. The data contains a unique ID named 'Run'. **Merged OTU data will be saved in [mergedData] directory.**

3. makeMergedMetadata.R

Description

No function for this process. Cleansing the metadata is always depends on user. But user should follow the specific direction below.

- From the metadata of each study, the user must extract and cleanse the meta information you need. (Unique ID for sample must be always included with named 'Run')
- The cleansed metadata of each study should be merged into one dataset.
- **Merged metadata should be saved in [mergedData] directory.**

4. makeTotalData.R

matchWithSampleID

Description

Automatically find matched sample ID between merged data and merged OTU.

Extract the corresponding samples on each data set and some columns are removed which sum as zero.

Usage

```
matchWithSampleID(meta, otu, taxa="Genus")
```

Arguments

meta	Input merged meta data which cleanse by user Unique ID should be named ‘Run’
otu	Input merged OTU data which is already prepared by ‘OTU.table’ function Only Run column and OTU column is allowed
taxa	Specify the taxa level which was used to make otu

Value

Return the OTU data named totalOTU+ taxa level and metadata named totalMetaData.

Two Total data will be saved in [totalData] directory

5. findSubStratification.R

drNclusterARI

Description

Perform dimensionality reduction method, apply clustering method and evaluate ARI to see which variables (external factor) divide the OTU data well. This process is repeated by the parameter ‘rep’ designated by the user and visualizes each external factor once as necessary.

Usage

```
drNclusterARI(totalOTUGenus, totalMetaData, taxa="Genus", method=2, rep=2,
visual=FALSE)
```

Arguments

data	Input total OTU data. No other columns are allowed.
meta	Input total meta data.
method	Dimensional reduction method to use (1: Barnes-Hut-SNE, 2: UMAP)
rep	The number of ARI calculation for each external factors. i.e., re-extraction

	for the number of ARI samples
visual	a logical; Choose to visualize or not

Value

Return ARI table

hyposAssumption

Description

Perform homogeneity of variance test with Brown-Forsythe method, analysis of variance test with welch ANOVA or ANOVA, post hoc test with Dunnett method and visualize the difference of mean between groups. Finally, automatically extract covariates which p value is lower than 0.05 on post hoc result.

Usage

```
hyposAssumption(data=resampled, control="disease")
```

Arguments

data	Input ARI table data
control	Input control groups which you want to compare with all other groups. control should be one of the columns of the ARI table must be specified. In this study, you should assign ‘disease’

Value

Return covariates vector

6. glmModeling.R

regressWithCovaAdjmt

Description

Perform modeling with edgeR package. Covariates are included in model for adjustment and implement comparison test simultaneously.

Usage

```
regressWithCovaAdjmt(data=totalOTUGenus, meta=totalMetaData, covariates=
covariates)
```

Arguments

data	Input total OTU data
meta	Input total meta data
covariates	Input covariates as string vector in meta data. disease column must be located first (because of detecting full rank process), and rest of the column must be arranged by high influence meta data. Use 'hyposAssumption' function result as covariates

Value

Return the results below

LogCPM	Change OTU data to cpm based on TMM normalization
glmResult	Generalized linear model results
TotalFTest	Result of ANOVA F Test based on 'disease' variable
disease-VSothers	Result of contrast test based on specific disease VS other disease

7. classification. R

clftWithONOF

Description

The logCPM value returned from the glm modeling and the variable importance vector based on the Total F test are used as parameters. The optimal number of variables is selected by sequentially increasing the number of variables based on the importance of variables. The classification model is performed with the corresponding variable, and the sensitivity of each disease is measured and returned. A graph of the optimal number of variables and a graph of final classification performance is output.

Usage

```
clftWithONOF(data=genusModelResult$LogCPM, target=totalMetaData$disease,
featureImpt=genusModelResult$TotalFTest['F',drop=FALSE], method="kknn")
```

Arguments

data	Input normalized OTU data which is returned by 'regressWithCovaAdjmt'
target	Input target variable from meta data. (Disease column)
featureImpt	Input feature importance based on Total F test which is returned by 'regressWithCovaAdjmt'

method Input classification method which used on caret package.

Value

Return the performance of specified classification