

Relate Mashup net weights to GeneMANIA

Data

Input: Mashup and genemania example shared same input.

- TCGA Breast cancer dataset. Information used was patient ID and whether tumour is of subtype 'Luminal A' (LumA) or other.
- N=232 patients. Classes={**LumA**, **other**} annotation.
- Similarity nets defined at the level of pathways, using Pearson correlation (ProfileToNetworkDriver) as similarity. Generates 1801 networks

Methods

How this variation of mashup algorithm works

1. Random walk with restarts: For each similarity network A_i , we run random walk to obtain Q_i .
2. Smoothing: For each Q_i , we select smooth or not to smooth. Smooth means $R_i = \log(\text{abs}(Q) + 1/n_{\text{patients}})$.
3. Concat each Q_i (not smooth) or R_i (smooth) along row axis to get a big matrix N .
4. Run SVD. $U, S, V = \text{svd}(N)$ to get U, S, V
 - a. Let $H = U * S^{(1/2)}$.
 - b. $V = S^{(1/2)} * V$.
5. Linear regression to get $\text{Beta} = V' \setminus \text{annotation}$ (left divide, annotation is a binary vector, 1 indicates query type, -1 indicates not-query type).

Getting network weights through cross-validation

```
For each class c1 in {LumA, other}
  Net_weights = matrix[N networks, 10 CV]
  For k in 1:10 # 10-fold CV
    Qry_k = 90% training samples of class c1
    For network j
      # indices corresponding to network j and
      # within that, samples in Qry_k
      H_cur = H[(j, Qry_k), ]
      All_weights = corr(H_cur, beta)
      # Or we can get network weight by dot product
      # All_weights = H_cur * beta
      Net_weights[j, k] = mean(All_weights)
    End
  End
End
```

End

Results

- Use LumA or other type as query, Run Smooth or not in the mashup. So we have $2 \times 2 = 4$ situation.
- For each situation, we have 10 query, so we have 10 figure. For each figure, one point represent a network, x axis is network weight from mashup, y axis is network weight from genemania.
- From figure title, we can have correlation and p-value between mashup weights and genemania weights.

Figures

1. {LumA, not smooth, correlation}

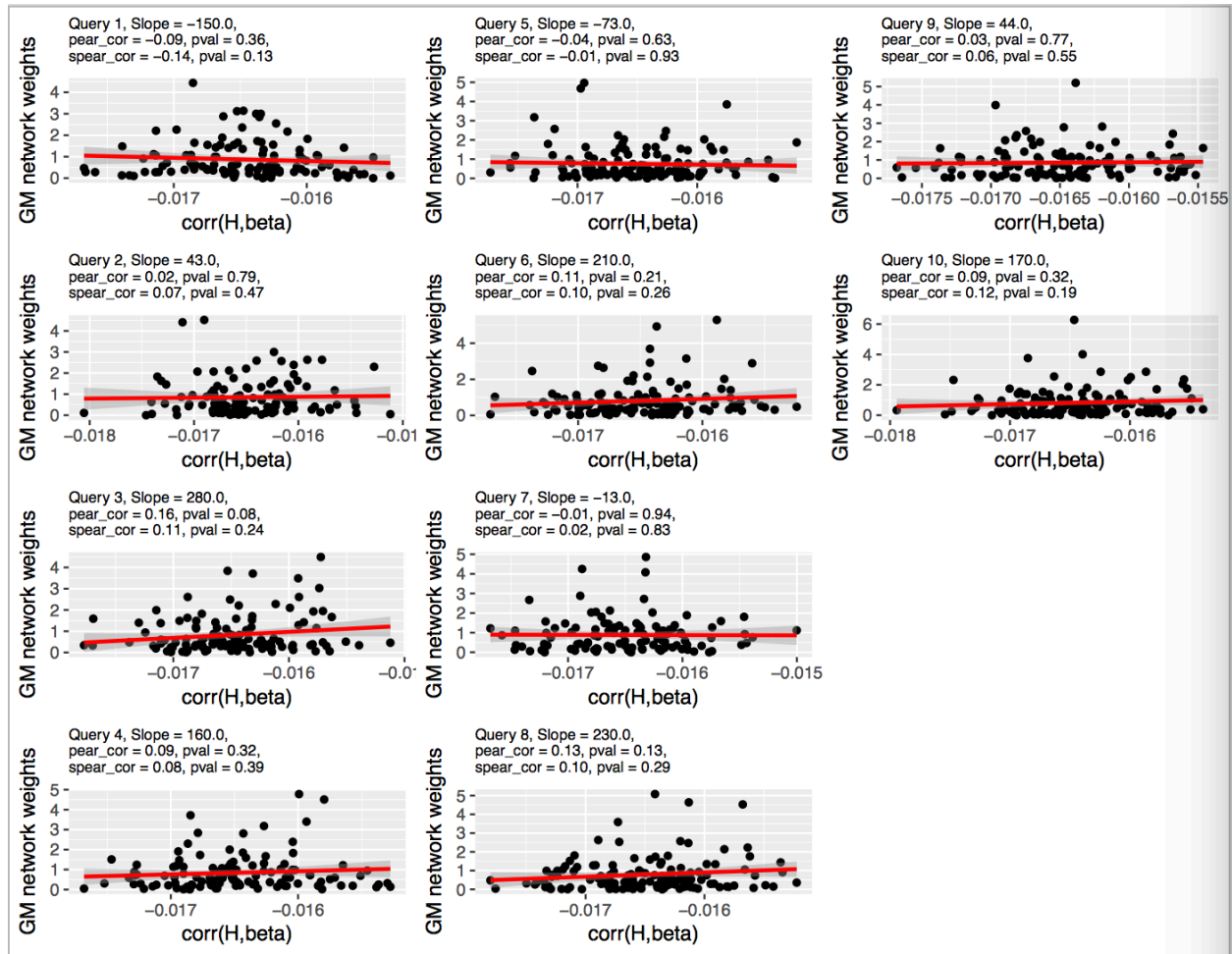


Figure 1. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

2. {LumA, not smooth, dot product}

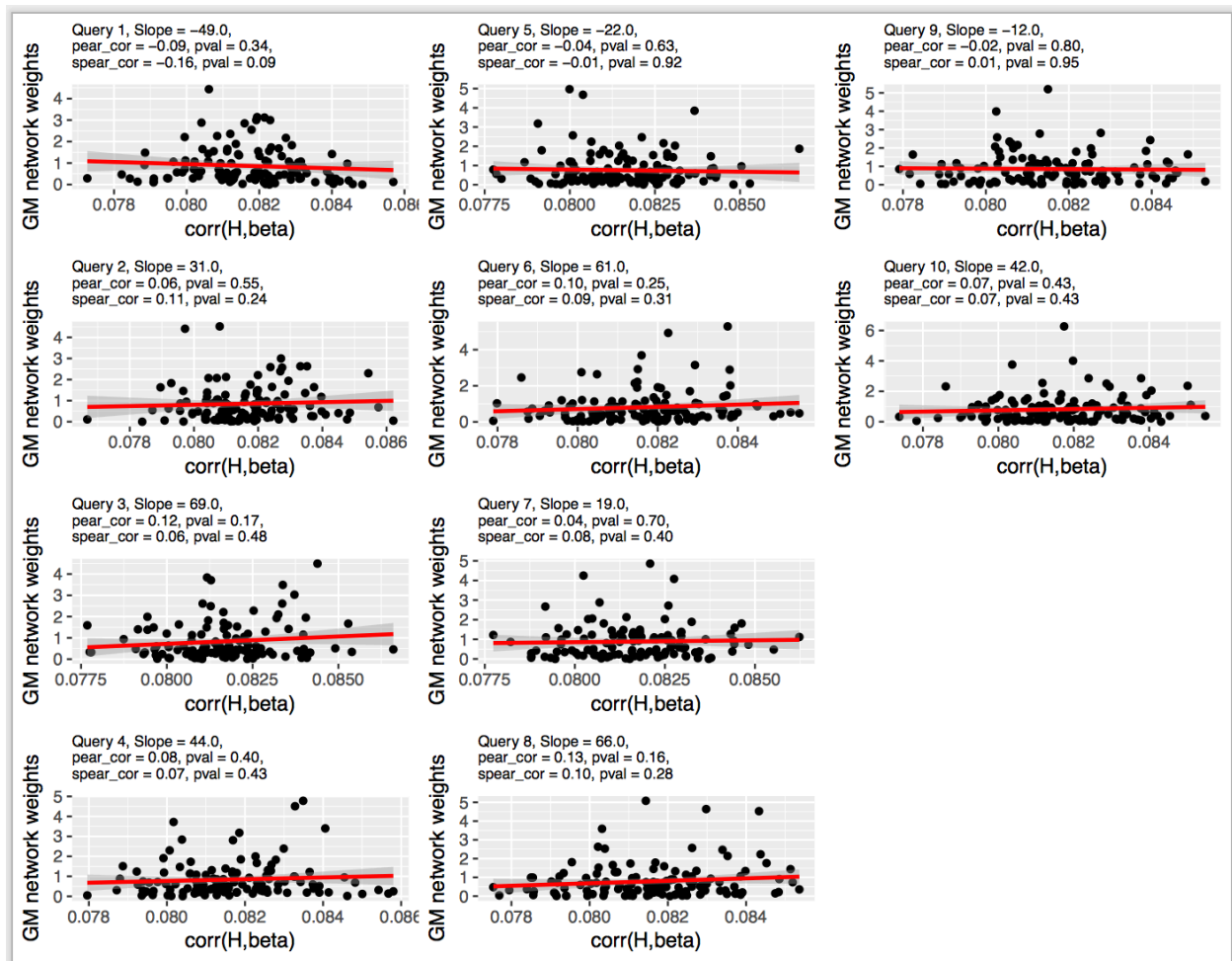


Figure 2. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

3. {LumA, smooth, correlation}

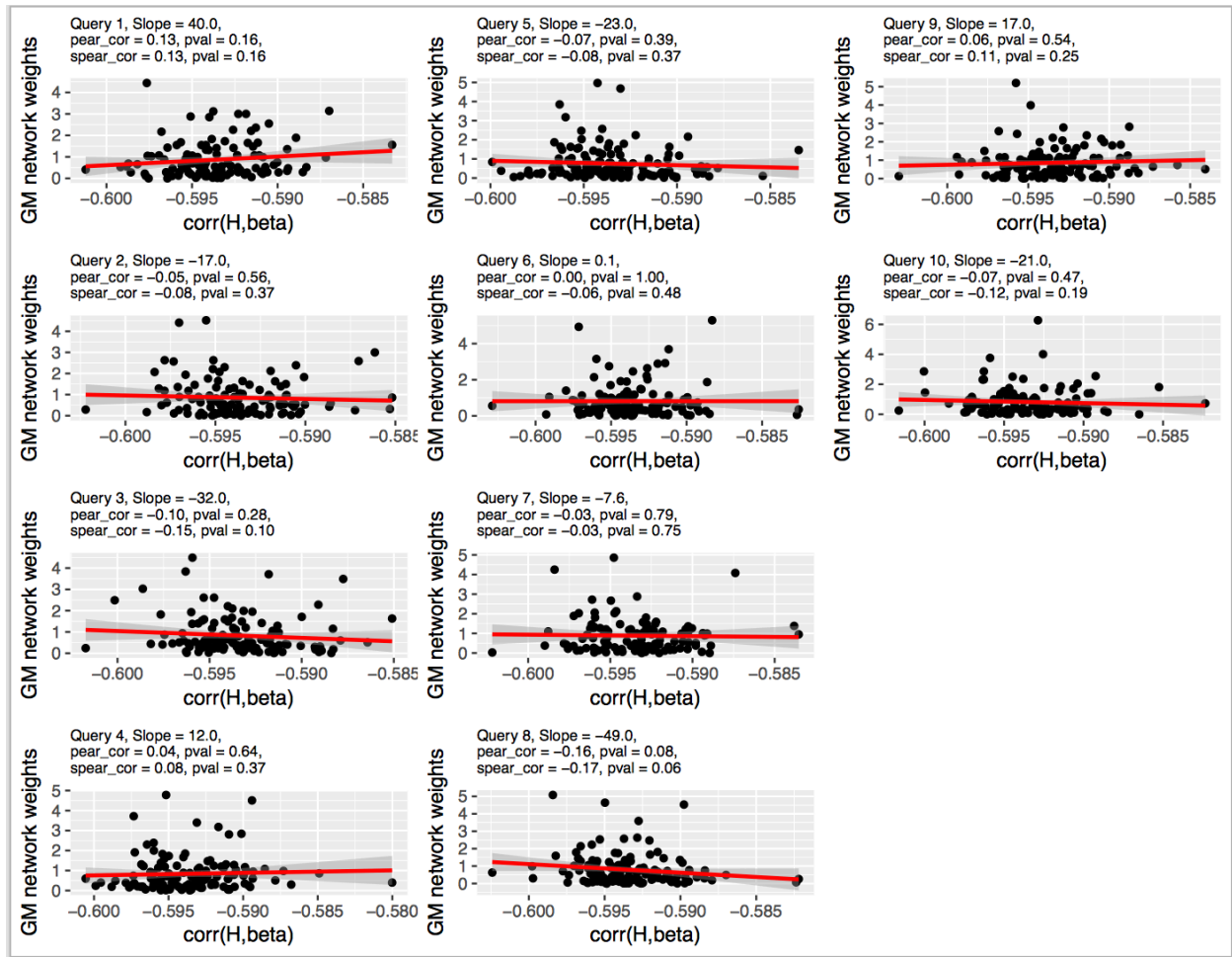


Figure 3. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

4. {LumA, smooth, dot product}

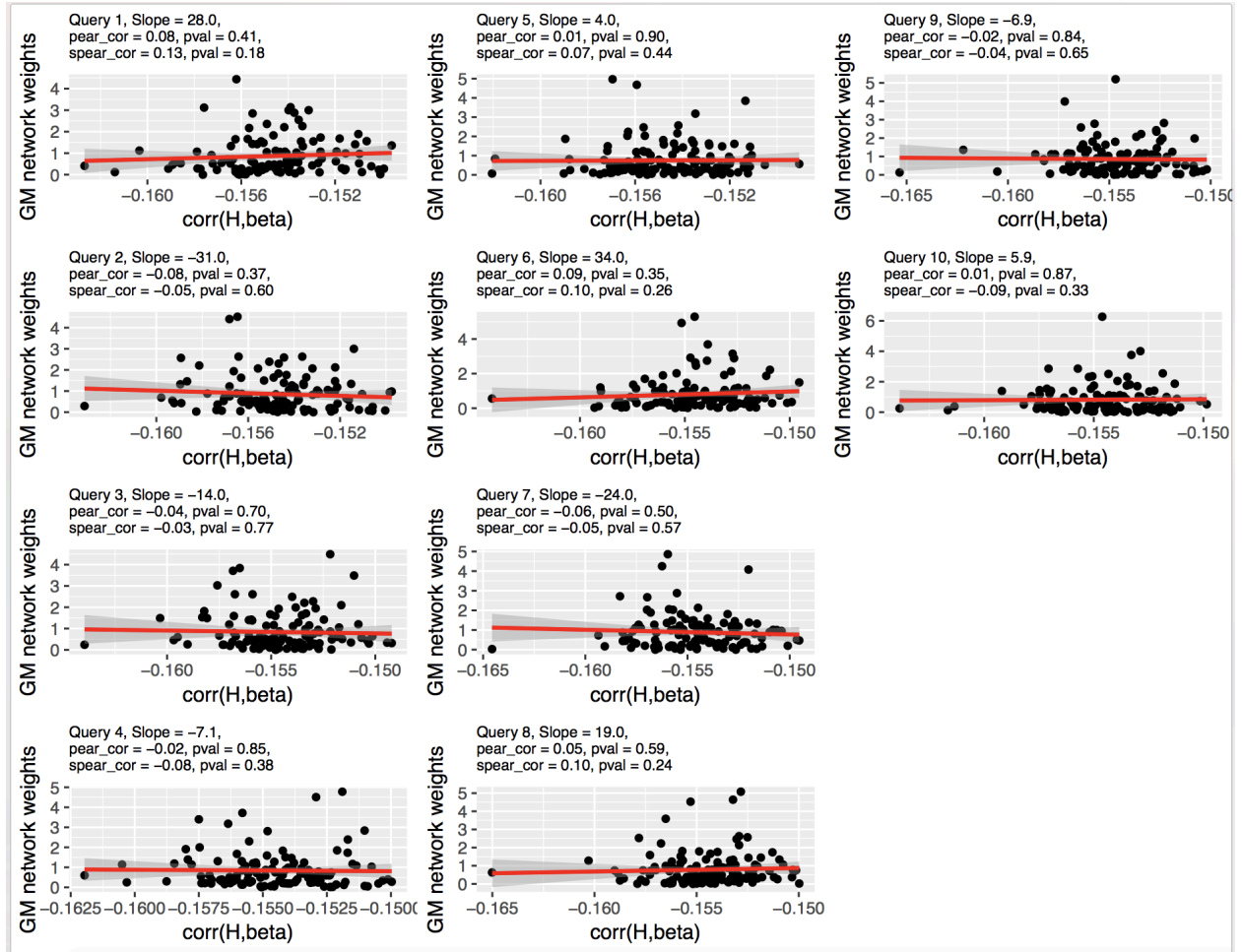


Figure 4. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

5. {other, not smooth, correlation}

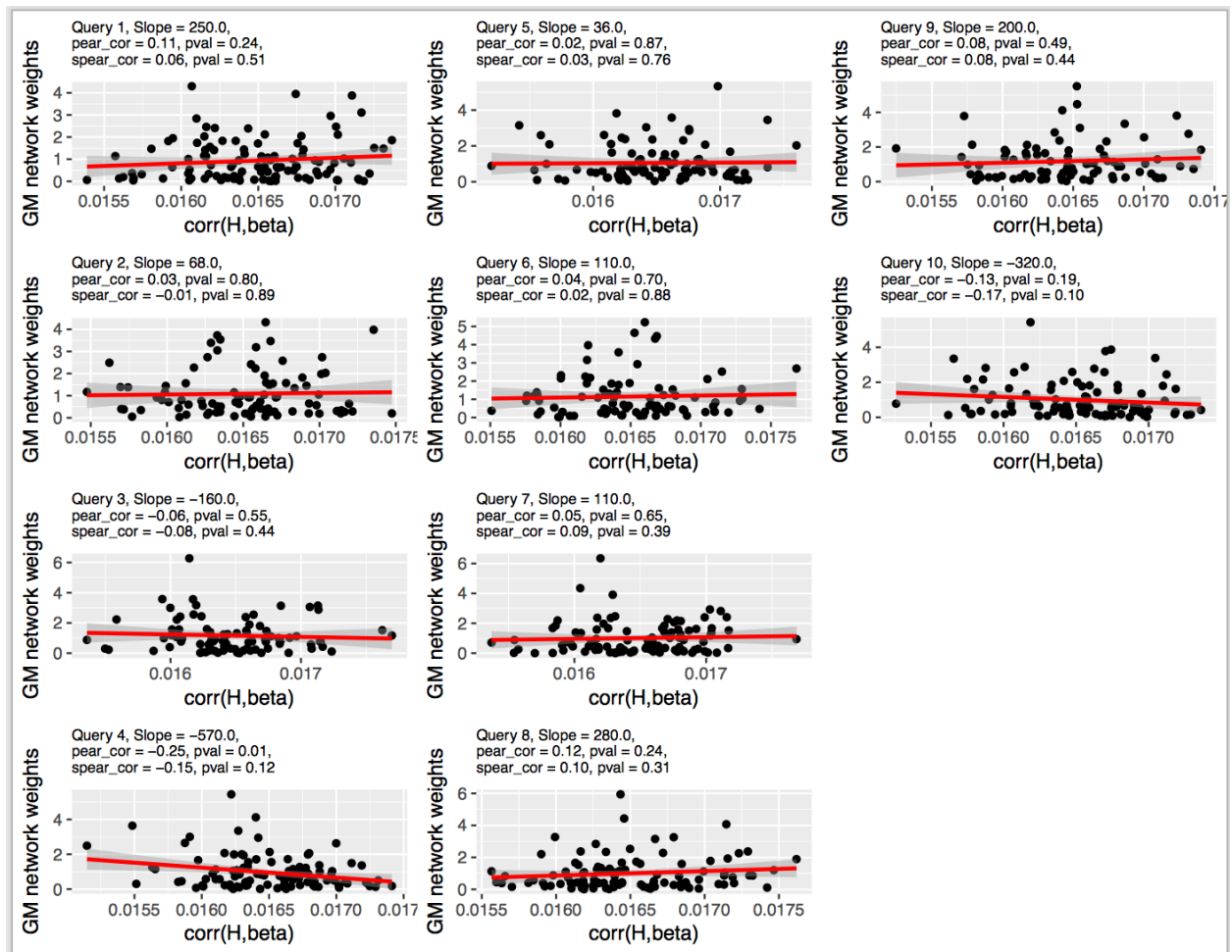


Figure 5. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

6. {other, not smooth, dot product }

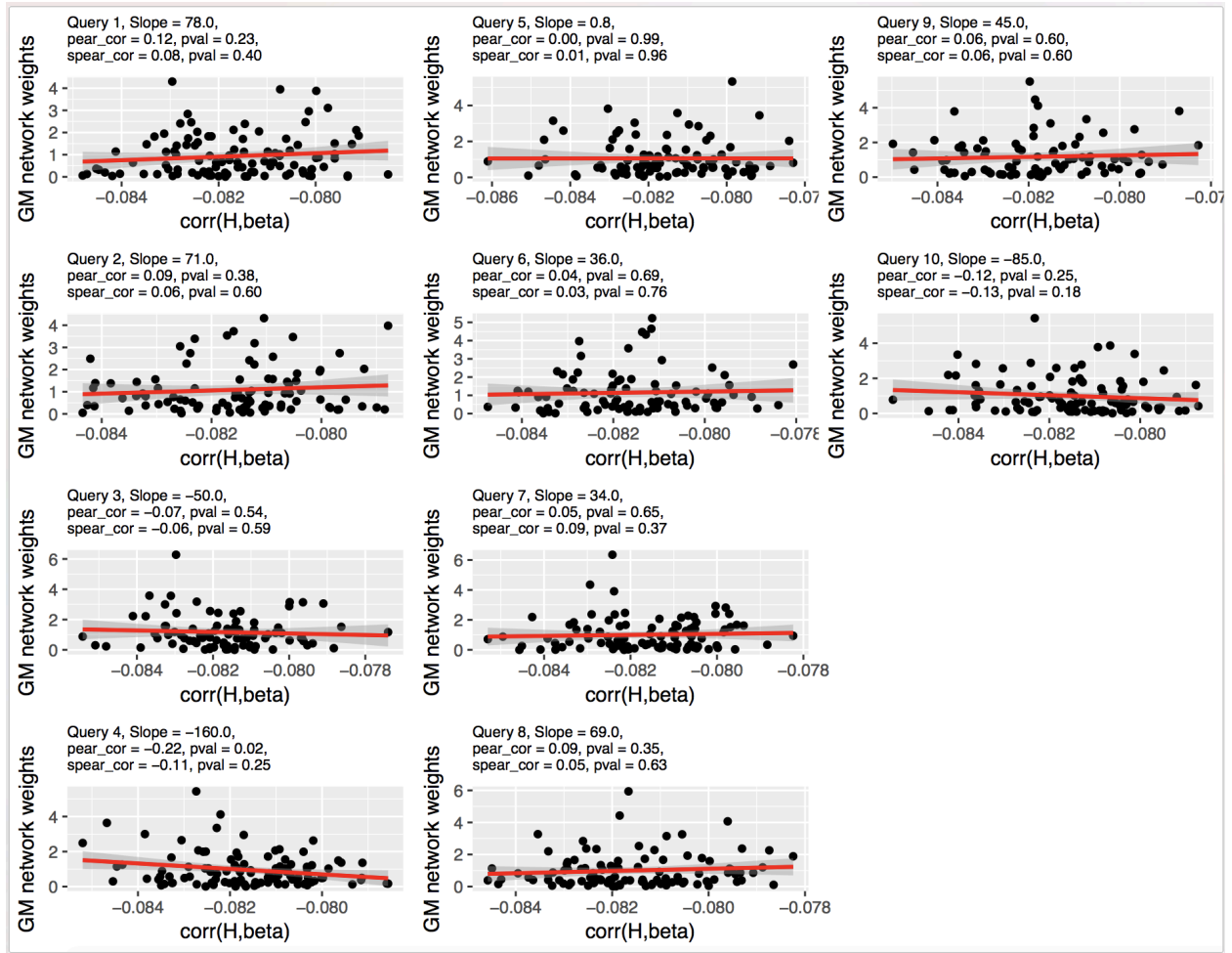


Figure 6. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

7. {other, smooth, correlation}

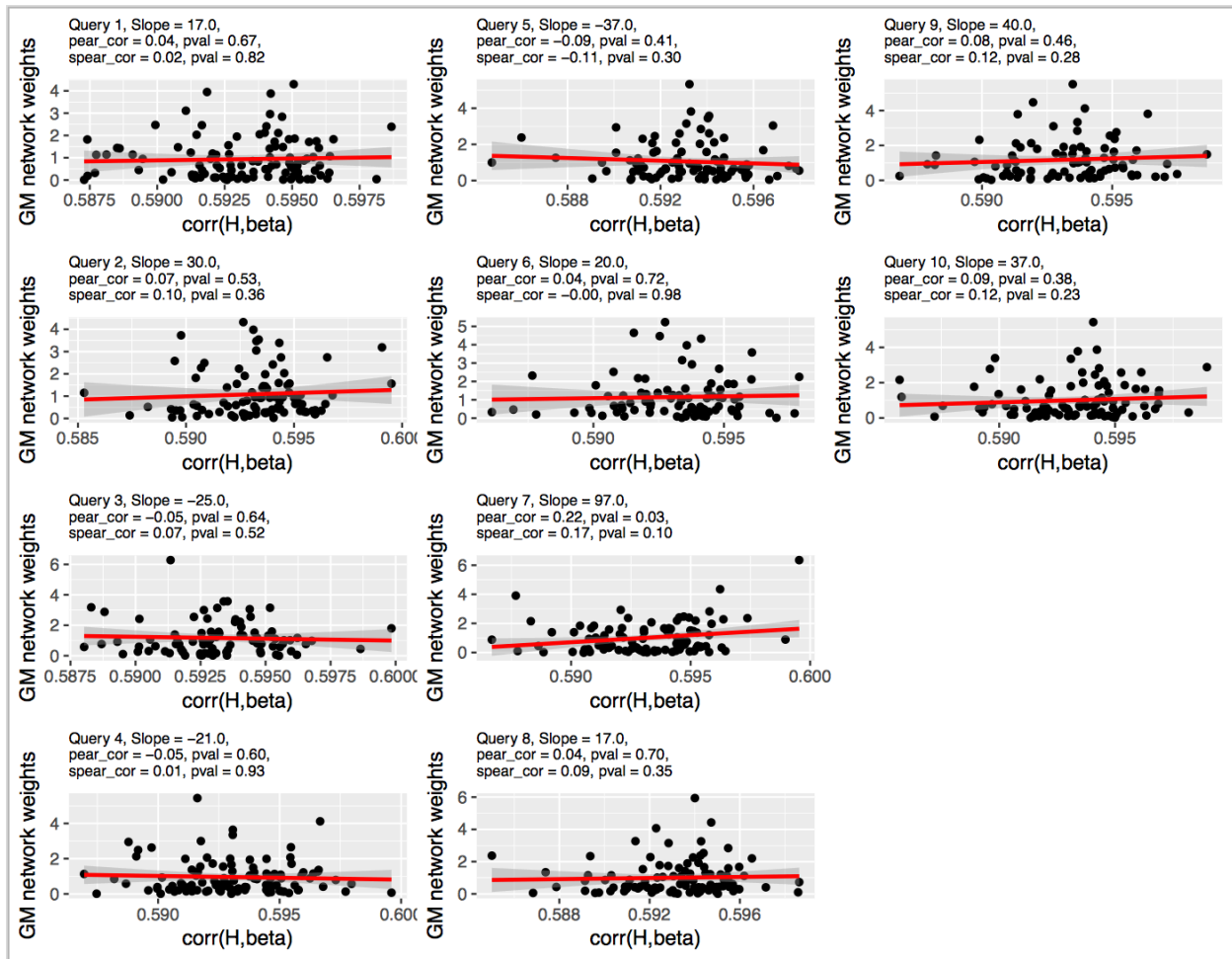


Figure 7. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

8. {other, smooth, dot product}

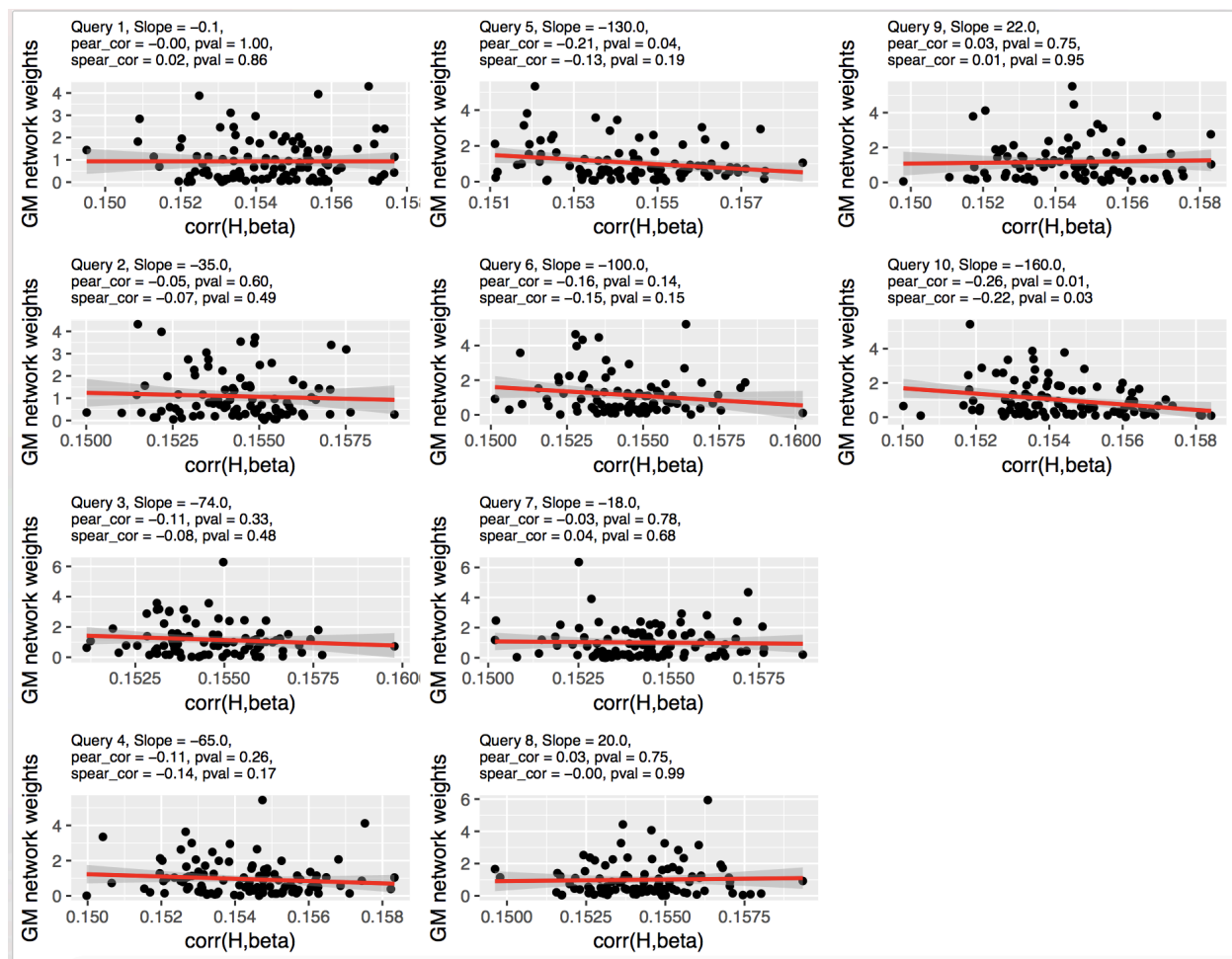


Figure 8. Network-level weights from Mashup (x-axis) and GeneMANIA (y-axis).

Tables

1. {LumA, not smooth, correlation}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	-0.09	-0.14
CV 2	0.02	0.07
CV 3	0.16	0.11
CV 4	0.09	0.08
CV 5	-0.04	-0.01
CV6	0.11	0.10
CV7	-0.01	0.02
CV8	0.13	0.10
CV9	0.03	0.06
CV10	0.09	0.12

Table 1. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

2. {LumA, not smooth, dot product}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	-0.09	-0.16
CV 2	0.06	0.11
CV 3	0.12	0.06
CV 4	0.08	0.07
CV 5	-0.04	-0.01
CV6	0.10	0.09

CV7	0.04	0.08
CV8	0.13	0.10
CV9	-0.02	0.01
CV10	0.07	0.07

Table 2. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

3. {LumA, smooth, correlation}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.13	0.13
CV 2	-0.05	-0.08
CV 3	-0.10	-0.15
CV 4	0.04	0.08
CV 5	-0.07	-0.08
CV6	0.00	-0.06
CV7	-0.03	-0.03
CV8	-0.16	-0.17
CV9	0.06	0.11
CV10	-0.07	-0.12

Table 3. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

4. {LumA, smooth, , dot product}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.08	0.13
CV 2	-0.08	-0.05
CV 3	-0.04	-0.03
CV 4	-0.02	-0.08
CV 5	0.01	0.07
CV6	0.09	0.10
CV7	-0.06	-0.05
CV8	0.05	0.10
CV9	-0.02	-0.04
CV10	0.01	-0.09

Table 4. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

5. {other, not smooth, correlation}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.11	0.06
CV 2	0.03	-0.01
CV 3	-0.06	-0.08
CV 4	-0.25	-0.15
CV 5	0.02	0.03
CV6	0.04	0.02
CV7	0.05	0.09
CV8	0.12	0.010
CV9	0.08	0.08

CV10	-0.13	-0.17
------	-------	-------

Table 5. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

6. {other, not smooth, , dot product}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.12	0.08
CV 2	0.09	0.06
CV 3	-0.07	0.06
CV 4	-0.22	-0.11
CV 5	0.00	0.01
CV6	0.04	0.03
CV7	0.05	0.09
CV8	0.09	0.05
CV9	0.06	0.06
CV10	-0.12	-0.13

Table 6. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

7. {other, smooth, correlation}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.04	0.02
CV 2	0.07	0.10
CV 3	-0.05	0.07
CV 4	-0.05	0.01

CV 5	-0.09	-0.11
CV6	0.04	-0.00
CV7	0.22	0.17
CV8	0.04	0.09
CV9	0.08	0.12
CV10	0.09	0.12

Table 7. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

8. {other, smooth, dot product}

CV ROUND	cor(mashup, GM), pearson	cor(mashup, GM), Spearman
CV 1	0.00	0.02
CV 2	-0.05	-0.07
CV 3	-0.11	-0.08
CV 4	-0.11	-0.14
CV 5	-0.21	-0.14
CV6	-0.16	-0.15
CV7	-0.03	0.04
CV8	0.03	0.00
CV9	0.03	0.01
CV10	-0.16	-0.22

Table 8. Correlation between Network-level weights from Mashup and GeneMANIA (Red indicates p-value < 0.1)

Summary Table

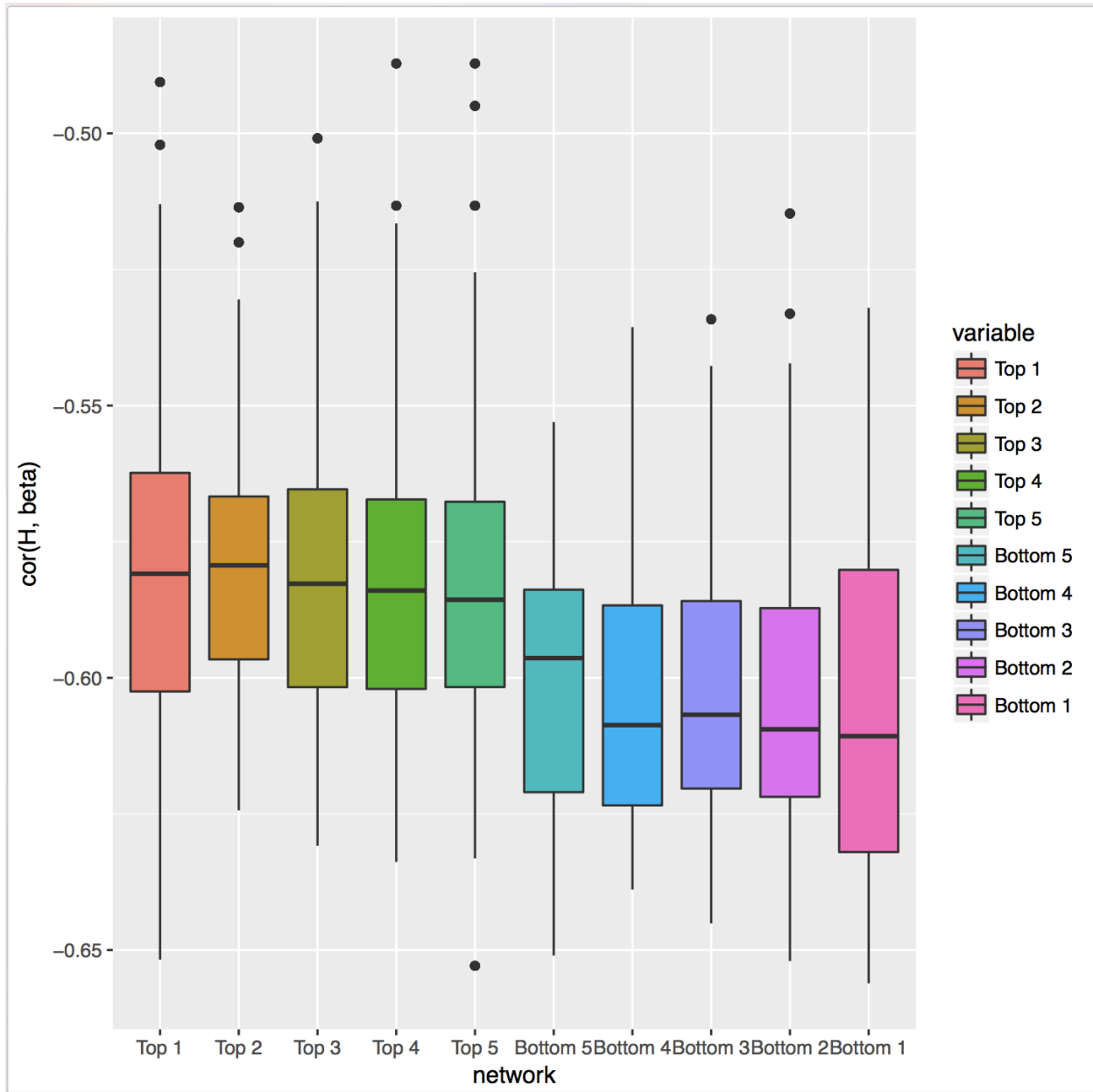
	pearson_cor	spearman_cor
{LumA, not smooth, correlation}	0.049	0.051
{LumA, not smooth, dot product}	0.045	0.042
{LumA, smooth, correlation}	-0.025	-0.037
{LumA, smooth, dot product}	0.002	0.006
{other, not smooth, correlation}	0.001	-0.012
{other, not smooth, dot product}	0.004	0.02
{other, smooth, correlation}	0.039	0.059
{other, smooth, dot product}	-0.077	-0.073

Table 5. Average Correlation between Network-level weights from Mashup and GeneMANIA
(Red indicates p-value < 0.1)

We can see that with not smooth, average correlation could be always positive.

Task 2: The distribution of $\text{corr}(H, \beta)$.

Here I explore the element-wise distribution of $\text{corr}(H, \beta)$ for 10 networks in {LumA, smooth}. Take top high-scoring and low-scoring networks from mashup(say 5 of each = 10 networks).



Task 3: Effect of smoothing before SVD

type	Num dims needed to explain 70% variation	Num dims needed to explain 80% variation	Num dims needed to explain 90% variation	Mean Pearson correlation between corr(H,beta) and GM	Mean Pearson correlation p-value between corr(H,beta) and GM
{LumA, not smooth}	157 dims (67% of total)	182 dims (78% of total)	207 dims (89% of total)	0.049	0.051
{LumA, smooth}	1 dims (0% of total)	1 dims (0% of total)	1 dims (0% of total)	-0.025	-0.037
{other, not smooth}	157 dims (67% of total)	182 dims (78% of total)	207 dims (89% of total)	0.001	-0.012
{other, smooth}	1 dims (0% of total)	1 dims (0% of total)	1 dims (0% of total)	0.039	0.059

Question

1. Linear regression for beta is not much precise. For example, we want $V * \beta = \text{disease}(-1, 1 \text{ vector})$, so I just do a simple linear regression to get beta, which is $\beta = \text{inv}(V) * \text{disease}$. Here I checked the accuracy of the beta, get: $\text{sum}(\text{sign}(V * \beta) == \text{disease})$, which is 78, however we have 232 sample, so linear regression only get $(232-78)/232 = 66\%$ accuracy. Hence beta is not what we exactly want, the question is **Will that put a negative impact on the estimation of the network weight?**
2. Another question is the way we obtain network weights from mashup. For example, for a specific network and a query, we locate its location at H(one of the rows) , then we compute the correlation between that row and beta to represent network weights. Then the question is that **can it represent network weight just like genemania want? Can we have another strategy to get network weight?** As you can see from the report, the result is not good as expected. What's more, I actually also have tried to use cross product between H and beta, the result is also not good and nearly same with correlation.