Accounting for unobserved factors when testing RNA-seq data for differential expression

Laurel Stell

March 6, 2019

Biostatistics consulting

- Data Studio
  - Every Wednesday 1:30-3:00 during fall, winter, spring quarters
  - Occasionally a presentation like this one
  - Drop-in consulting once a month
  - Usually an in-depth consultation for a Medical School researcher
    - Want to present your project?
    - Want to learn about other SOM projects?
    - Want to see biostatisticians in action?
- Individual consultations
- Brought to you by Spectrum and DBDS

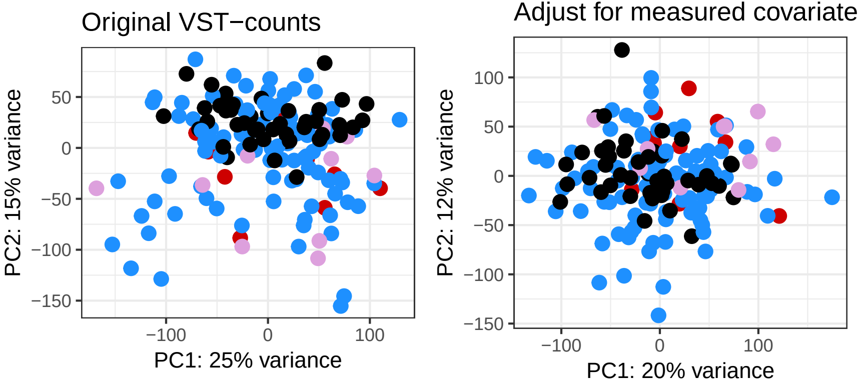
med.stanford.edu/dbds/cool-tools/data-studio.html

More info, including how to get Data Studio announcements or request a consultation

Example

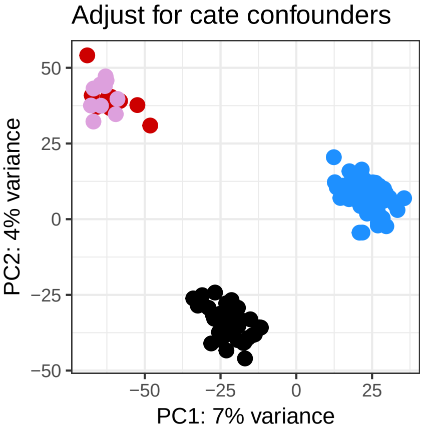
- 161 human blood samples sequenced
- 3 disease groups: controls, first-degree relatives of T1D patient, AA+
- Other measured covariates:
  - Sex
  - Age
  - Sample collection site
  - RNA processing site
  - RNA extraction kit
  - Sequencing batch
  - RIN

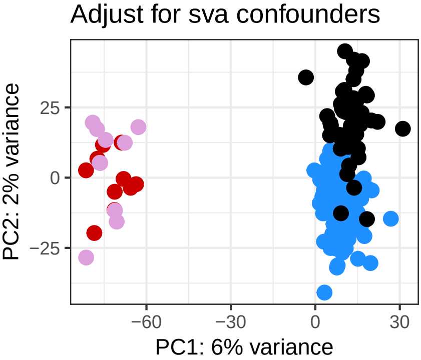
PCA plots



Original VST−counts
PC2: 15% variance
50
0
−50
−100
−150
−100
0
100
PC1: 25% variance
Adjust for measured covariate
PC2: 12% variance
100
50
0
−50
−100
−150
−100
0
100
PC1: 20% variance

PCA plots (continued)





Outline

- Mathematical model
- Estimating number of latent confounders
- Estimating confounders with cate or sva
- Testing for differential expression (DE)
- More comparisons of results

The model

N samples, G genes, K unmeasured confounders; $K \ll N \ll G$

$$\mathbf{Y}_{N\times G} = \mathbf{X}_{N\times 1}\beta^T_{G\times 1} + \mathbf{Z}_{N\times K}\mathbf{\Gamma}^T_{G\times K} + \mathbf{E}_{N\times G}$$

$$\mathbf{Z} = \mathbf{X}\alpha^T_{K\times 1} + \mathbf{W}_{N\times K}$$

- $\mathbf{Y}$ contains RNA-seq measurements suitably massaged
- $\mathbf{X}$ contains variable being tested for association. Can add more measured covariates, both primary and nuisance parameters.
- $\mathbf{E}$ is noise with iid rows: $\mathbf{E}_i \sim N(0, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = \mathrm{diag}(\sigma^2_1, \dots, \sigma^2_G)$
- $\mathbf{W}$ has iid standard normal entries
  - Not just noise
  - Needs to be a substantial part of $\mathbf{Z}$ or it will be impossible to solve for β
- $\mathbf{W} \perp\!\!\!\perp \mathbf{X}$ and $\mathbf{E} \perp\!\!\!\perp (\mathbf{X}, \mathbf{Z})$

Goal: Test G hypotheses $H_g : \beta_g = 0$

Difficulty: $\mathbf{Z}$ is unknown

Multiple regression version

$$\mathbf{Y} = \mathbf{X}_0\mathbf{B}_0^T + \mathbf{X}_1\mathbf{B}_1^T + \mathbf{Z}\mathbf{\Gamma}^T + \mathbf{E}$$

$$\mathbf{Z} = \mathbf{X}_0\mathbf{A}_0^T + \mathbf{X}_1\mathbf{A}_1^T + \mathbf{W}$$

- $\mathbf{X}_1$ contains d_1 covariates of interest
- $\mathbf{X}_0$ contains d_0 nuisance covariates

Variance stabilizing transformation

- RNA-seq counts for a gene typically have variance that increases with the average
- PCA plots and above model require homoscedasticity
- log-transformation over-corrects genes with low expression
- vst() in DESeq2 gives roughly log2-transformation for highly-expressed genes but better for other genes
- rlog() in DESeq2 is recommended, but it takes too long if you have very many samples

What about FPKM?

I only have FPKM from Cufflinks. What should I do?

- Many benchmarks show that FPKM is not a good normalization approach:
    - Dillies et al. (2012)
    - Seyednasrollah, Laiho and Elo (2013)
    - Rapaport et al. (2013)
    - Zhang et al. (2014)
    - Schurch et al. (2016)
- Lior Pachter, whose lab developed Cufflinks, says not to use it
    - pachterlab.github.io lists it under “retired software”
    - In CSHL keynote in 2013 (www.youtube.com/watch?v=5NiFibnbE8o about 35 minutes in), explains that it is wrong:
        - FPKM discards a proportionality constant
        - That constant differs between experiments
        - So FPKM is not appropriate for DE testing

Simple approaches to handle latent factors

- Naive: Ignore and just do linear regression of $\mathbf{Y}$ on $\mathbf{X}$
  - Gives unbiased estimate for $\beta + \mathbf{\Gamma}\alpha$
  - Fine if $\alpha = 0$, ie $\mathbf{X} \perp\!\!\!\perp \mathbf{Z}$
- GWAS: Remove leading principal components from $\mathbf{Y}$
  - Fine if confounder signal much stronger than effect of primary variable
  - Otherwise, gives biased estimate of $\mathbf{Z}$
- Estimate $\widehat{\mathbf{Z}}$ by leading principal components of residual matrix from naive approach
  - Actually estimates $\mathbf{W}$
  - Again, fine if $\mathbf{X} \perp\!\!\!\perp \mathbf{Z}$
  - OLS of $\mathbf{Y}$ on $(\mathbf{X}\ \widehat{\mathbf{Z}})$ will give same estimate of $\mathbf{B}$ as naive approach

cate rotation (Wang, Zhao, Hastie and Owen, 2017)

- $N \times N$ Householder matrix $\mathbf{Q}^T$ such that $\mathbf{Q}^T\mathbf{X} = \|\mathbf{X}\|_2\mathbf{e}_1$

$$\widetilde{\boldsymbol{Y}} = \mathbf{Q}^T\mathbf{Y} = \|\mathbf{X}\|_2\mathbf{e}_1\beta^T + \widetilde{\boldsymbol{Z}}\boldsymbol{\Gamma}^T + \widetilde{\boldsymbol{E}}$$

$$\widetilde{\boldsymbol{Z}} = \mathbf{Q}^T\mathbf{Z} = \|\mathbf{X}\|_2\mathbf{e}_1\alpha^T + \widetilde{\boldsymbol{W}}$$

- Rotation does not change distributions of $\mathbf{E}$ or $\mathbf{W}$
- Separating first row from the rest:

$\widetilde{\boldsymbol{Y}}_1 = \|\mathbf{X}\|_2\beta^T + \widetilde{\boldsymbol{Z}}_1\boldsymbol{\Gamma}^T + \widetilde{\boldsymbol{E}}_1 \sim N(\|\mathbf{X}\|_2(\beta + \boldsymbol{\Gamma}\alpha)^T, \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T + \boldsymbol{\Sigma})$

$\widetilde{\boldsymbol{Y}}_{-1} = \widetilde{\boldsymbol{Z}}_{-1}\boldsymbol{\Gamma}^T + \widetilde{\boldsymbol{E}}_{-1}$ has rows iid $N(0, \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T + \boldsymbol{\Sigma})$, which is exploratory factor analysis model

Estimating Number of Confounders

Factor analysis model

$$\boldsymbol{\Upsilon} = \boldsymbol{\Lambda} + \boldsymbol{\Xi}\boldsymbol{\Sigma}^{1/2} = (\sqrt{N}\mathbf{U}_{N\times K}\mathbf{D}\mathbf{V}_{G\times K}^{T} + \boldsymbol{\Xi})\boldsymbol{\Sigma}^{1/2}$$

- $\boldsymbol{\Lambda}$ has rank K
- $\boldsymbol{\Sigma}$ same as above
  - Heteroscedastic noise
  - $\boldsymbol{\Sigma} = \sigma I_G$ is homoscedastic or white noise
- $\boldsymbol{\Xi}$ has iid entries with mean 0 and variance 1
- $\mathbf{U}, \mathbf{V}$ are orthogonal
- $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_K)$

Goal: Compute $\widehat{\boldsymbol{\Lambda}}$ that minimizes loss function $\mathrm{Err}_{\Lambda}(\widehat{\boldsymbol{\Lambda}}) \equiv \mathbb{E}(\left\|\widehat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}\right\|_F^2)$.

SVD in white noise (Perry, Stanford dissertation, 2009)

Assume $\mathbf{\Sigma} = I_G$ and $N, G \to \infty$ with $G/N \to \gamma$. RNA-seq has large γ (at least 50, say).

- $\|\mathbf{\Xi}\|_F^2 \approx NG$
- $\mathbf{\Lambda} = \sqrt{N}\mathbf{UDV}^T$
    - $\mathbf{\Lambda}^T\mathbf{\Lambda}/N$ has eigenvalues $\mu_k = d_k^2$
    - Each factor contributes $\|\mathbf{\Upsilon}_k\|_F^2 = N\mu_k$
    - $\mu_k > G$ for $\mathbf{\Upsilon}_k$ to exceed noise

SVD in white noise (part 2)

- If first K singular values of $\boldsymbol{\Upsilon}$ are $\sqrt{N\widehat{\mu}_k}$, then $\widehat{\mu}_k \to \widetilde{\mu}_k$, where

$$\widetilde{\mu}_k = \begin{cases} (\mu_k + 1)\left(1 + \frac{\gamma}{\mu_k}\right) & \text{when } \mu_k > \sqrt{\gamma} \\ (1 + \sqrt{\gamma})^2 & \text{otherwise} \end{cases}$$

  If any $\mu_k < \sqrt{\gamma}$, then corresponding eigenvalue of $\boldsymbol{\Upsilon}^T\boldsymbol{\Upsilon}/N$ should be near λ_+ from Marchenko-Pastur law.
- Including factor k in $\widehat{\boldsymbol{\Lambda}}$ would increase loss if μ_k less than roughly γ for large γ
- Invalidates EstDimRMT in isva package, as well as counting eigenvalues > 1

SVD in white noise (part 3)

4 categories of factors:

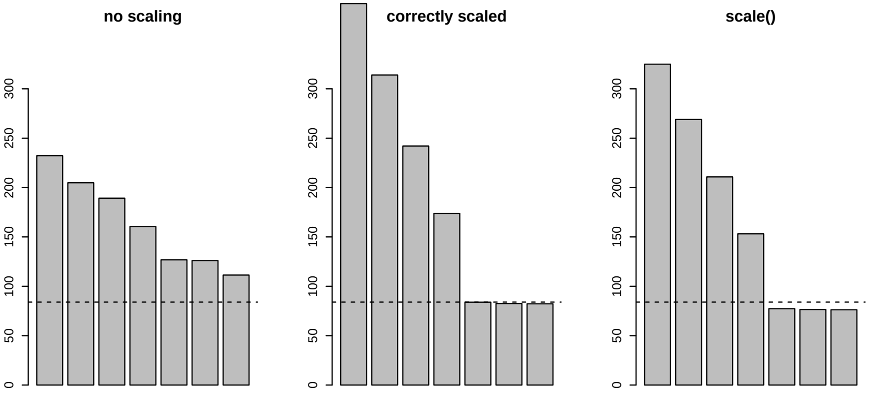
Undetectable: $\mu_k < \sqrt{\gamma}$ implies factor undetectable by SVD-based methods

Harmful: $\sqrt{\gamma} < \mu_k < \mu_F^*$ implies $\widehat{\mu}_k > \lambda_+$ without doubt, but including factor in $\widehat{\mathbf{\Lambda}}$ increases loss

Useful: $\mu_k > \mu_F^*$ but still $O(1)$ implies including factor will decrease the loss but factor contribution still smaller than noise

Strong: $\mu_k \sim O(G)$ implies factor larger than noise and $\mathbf{U}_{*k}$ can be estimated very well

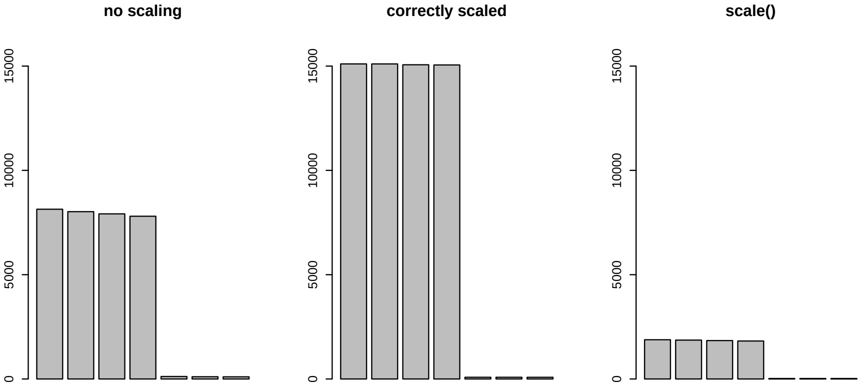
Heteroscedasticity (Owen and Wang, 2016)



Scree plots ($\widehat{\mu}_k$) for simulated heteroscedastic data with 4 useful weak factors for eigenvalues based on $\boldsymbol{\Upsilon}$, $\boldsymbol{\Upsilon}\boldsymbol{\Sigma}^{-1/2}$, and $\textbf{scale}(\boldsymbol{\Upsilon})$. Without scaling, scree plot does not reveal how many factors there are.

Heteroscedasticity (part 2)

- No strong factors implies **E** dominates **Λ** and column variances $\widehat{\sigma}_g$ approximate σ_g^2 well
- With 4 strong factors, cor$(\widehat{\sigma}_g^2, \sigma_g^2) < 0.6$ and using **scale()** gives slightly greater loss than no scaling



Heteroscedasticity (part 3)

To emphasize

Do not scale genes to have equal variances when estimating number of confounding factors if there might be strong factors.

Further invalidates EstDimRMT algorithm . . . not to mention several aspects of its implementation

Parallel analysis via permutation (Buja and Eyuboglu, 1992)

- $N \to \infty$ with G fixed
- Algorithm:
  - Compute nonzero singular values ν_k of $\boldsymbol{\Upsilon}$
  - $T_k = \nu_k^2 / \sum_\ell \nu_\ell^2$
  - For $b = 1, \dots, B$ ($B = 20$ by default)
    - Rearrange each column of $\boldsymbol{\Upsilon}$ independently
    - Compute $T_k^{(b)}$ of permuted matrix as before
  - For each k, let p_k be fraction of b with $T_k^{(b)} \geq T_k$.
  - $K = \min_k\{k : p_k > \alpha\} - 1$, where $\alpha = 0.1$ is hard-wired
- $\sum_k \nu_k$ doesn't change between first step and permutations because:
  - Sum of eigenvalues equals trace
  - Rearranging entries in a column does not change diagonal of $\boldsymbol{\Upsilon}^T\boldsymbol{\Upsilon}$
- Weaknesses:
  - Misses useful weak factors when there are strong factors
  - In Owen and Wang tests, inflates $\widehat{K}$ when no strong factors and only one useful factor

Large matrices with strong factors

$N, G \to \infty$ with K fixed and $\mu_k \sim G$ or larger

- Owen and Wang tested several methods
- Best was sample covariance eigenvalue difference (ED) method of Onatski (2012)
    - $\widehat{K} = \max\{k \le K_{\max} : \lambda_k^2 - \lambda_{k+1}^2 \ge \delta\}$
    - Parameters $K_{\max}$ and δ chosen suitably
    - Only requires μ_k diverge in probability
    - Looks for gaps in eigenvalues, not for eigenvalues themselves to exceed given threshold

Large matrices with weak factors

$N, G \to \infty$ with K fixed and $\mu_k = O(1)$

- Attempt to estimate number of factors with $\mu_k > \sqrt{\gamma}$
- Owen and Wang tested information criterion of Nadakuditi and Edelman (2008)
- Results for white noise scenario show that:
  - This is hard because $\widehat{\mu}_k \sim \gamma$ or larger
  - Don't really want to use harmful factors anyway

Bi-cross-validation (Owen and Wang, 2016)

- Maximize log-likelihood for $\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}$
  - Actually unbounded if any $\widehat{\sigma}_g \to 0$
  - Use early stopping criterion of 3 iterations
- Randomly partition

$$\boldsymbol{\Upsilon} = \begin{pmatrix} \boldsymbol{\Upsilon}_{00} & \boldsymbol{\Upsilon}_{01} \\ \boldsymbol{\Upsilon}_{10} & \boldsymbol{\Upsilon}_{11} \end{pmatrix}$$

  and do clever linear algebra to estimate prediction error of $\boldsymbol{\Upsilon}_{00}$
- Use cross-validation to choose $\widehat{K}$ that minimizes CV PE
- Mathematically justified, not too hard to explain, implemented in cate package

Estimate Confounders Given $\widehat{K}$

Identifiability issues

- If $\mathbf{U}$ is orthogonal, then $\mathbf{ZUU}^T\mathbf{\Gamma}^T = \mathbf{Z\Gamma}^T$.
  - $\mathbf{Z}$ and $\mathbf{\Gamma}$ only determined up to a rotation
  - Sufficient to identify β
- For any $\mathbf{M} \in \mathbb{R}^{K \times d}$:

$$\mathbf{X}(\mathbf{B} + \mathbf{\Gamma M})^T + (\mathbf{Z} - \mathbf{XM}^T)\mathbf{\Gamma}^T = \mathbf{XB} + \mathbf{Z\Gamma}^T$$

Impossible to identify projection of β onto column space of $\mathbf{\Gamma}$. Common to use either of following:

  - Negative controls: Require known set $\mathcal{C}$ of K or more genes such that $\beta_{\mathcal{C}} = 0$ and $\text{rank}(\mathbf{\Gamma}_{\mathcal{C}}) = K$
  - Sparsity: $\|\beta\|_0 \leq \lfloor (G - S)/2 \rfloor$; $(\mathbf{\Gamma}_{\mathcal{C}}) = K$ if $|\mathcal{C}| = S$ for $K \leq S \leq G$

SVA (Leek & Storey, 2007, 2008)

Idea: If we knew a set $\mathcal{C}$ of genes such that $\beta_{\mathcal{C}} = 0$, then $\mathbf{Y}[,\mathcal{C}] = \mathbf{Z}\boldsymbol{\Gamma}_{\mathcal{C}}^{T} + \mathbf{E}[,\mathcal{C}]$ and PCA would give $\mathbf{Z}$.

- $\widehat{\mathbf{Z}}$ contains top K eigengenes in SVD of $\mathbf{R}$
- For $j = 1, \ldots, J$ ($J = 5$ by default)
  - Compute p-values for $\mathbf{B}_1$ in OLS fit of $(\mathbf{X}, \widehat{\mathbf{Z}})$
  - Compute p-values for $\boldsymbol{\Gamma}$ in OLS fit of $(\mathbf{X}_0, \widehat{\mathbf{Z}})$
  - Use Bayes theorem to estimate:
    - Probabilities P_X that $(\mathbf{B}_1)_g = 0$
    - Probabilities P_Z that $\boldsymbol{\Gamma}_g = 0$
  - $P = \widehat{P}_X(1 - \widehat{P}_Z)$
  - $\mathbf{R}_g^* = P_g \mathbf{R}_g$ and center columns
  - $\widehat{\mathbf{Z}}$ contains top K eigengenes in SVD of $\mathbf{R}^*$

SmartSVA (Chen et al, 2017)

- At end of each iteration, ρ is the Spearman correlation between new and previous values of P.
- Exits iteration when $1 - \rho < \epsilon$, which is 0.001 by default
- Maximum number of iterations is 100 by default
- Also uses QR decomposition of $\mathbf{X}$ to speedup computations.

Alas, this convergence metric does not measure how much $\widehat{\mathbf{Z}}$ is changing.

My convergence criterion

- Want to know whether the column space of $\widehat{\mathbf{Z}}$ is converging
- Let r be the maximum absolute value of the residuals from projecting columns of $\widehat{\mathbf{Z}}$ onto column space of estimate at previous iteration.

Alas, r often doesn't go to zero, so I used underrelaxation:

- If the algorithm using weights $\mathbf{w}_{j-1}$ at iteration j computes that the weights should be $\widehat{\mathbf{w}}$, instead set $\mathbf{w}_j = \mathbf{w}_{j-1} + \omega(\widehat{\mathbf{w}} - \mathbf{w}_{j-1})$
- $\omega \in (0, 1)$ chosen by trial but 0.5 usually works

cate

$\widetilde{\boldsymbol{Y}}_{-1} = \widetilde{\boldsymbol{Z}}_{-1}\boldsymbol{\Gamma}^T + \widetilde{\boldsymbol{E}}_{-1}$ has rows iid $N(0, \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T + \boldsymbol{\Sigma})$. Solve for $\boldsymbol{\Sigma}$ and $\boldsymbol{\Gamma}$.

Package implements 3 methods:

- Quasi-maximum likelihood (default)
- PCA
- Factor analysis using early stopping criterion of bi-cross-validation

DE Testing With Target False Discovery Rate 0.1

cate

$$\widetilde{\boldsymbol{Y}}_1 = \|\mathbf{X}\|_2\beta^T + \widetilde{\boldsymbol{Z}}_1\boldsymbol{\Gamma}^T + \widetilde{\boldsymbol{E}}_1 \sim N(\|\mathbf{X}\|_2(\beta + \boldsymbol{\Gamma}\boldsymbol{\alpha})^T, \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T + \boldsymbol{\Sigma})$$

Use estimated $\boldsymbol{\Sigma}$ and $\boldsymbol{\Gamma}$ for inference on $\boldsymbol{\alpha}$ and β

- Orthogonal rotation of $\boldsymbol{\Gamma}$ would also rotate $\boldsymbol{\alpha}$ but not affect β
- β and $\boldsymbol{\alpha}$ have $G + K$ parameters but $\widetilde{\boldsymbol{Y}}_1$ is length G, so not identifiable
- With sparsity assumption, cast problem as robust regression: nonzero entries in β correspond to outliers
- By default, calibrates t-statistics to account for difference between asymptotic results and finite sample size
- Rank genes by p-values

(Note: Without confounders, plain linear regression.)

limma

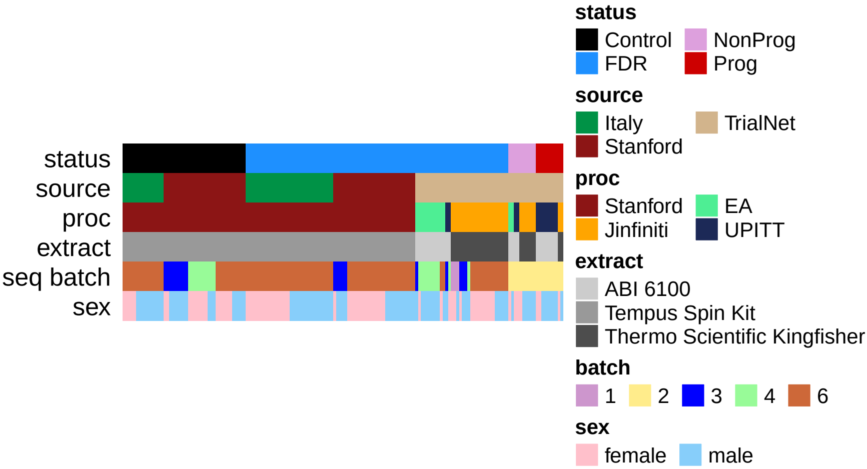
- Uses linear regression with Gaussian error but moderates the t-statistics
- Options for normalization (first two used below):
  - vst() as above
  - logCPM from edgeR and intensity-trend option when moderating t-statistics
  - voom if library sizes highly variable between samples (see chapter on RNA-seq in User's Guide)
- Uses B-statistic by default to rank genes
  - Log-odds that gene is differentially expressed
  - Will give same order as p-values if there are no missing values

DESeq2

- Negative binomial model of count data as output by:
    - RSEM
    - Pseudoalignment methods such as Kallisto
- In blog post, Love recommends using limma if there are 100s of samples because it's much faster.
- To rank genes:
    - DESeq2 documentation recommends ranking by shrunken |LFC| (using apeglm shrinkage)
    - Also rank by p-values in results below

Testing for DE Between AA+ and Controls

The samples



Test AA+ (both NonProg and Prog) versus controls

Mathematical model

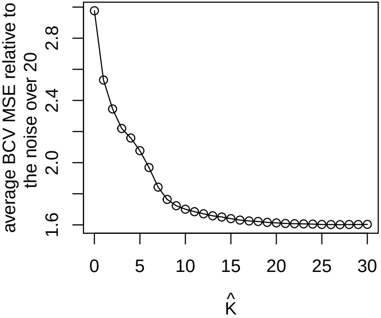
To avoid singular model matrix and improve its suitability:

- Discard extraction method because it is determined by processing site
- Combine collection site and processing site into single variable
- Combine batches 1, 2 and 3
- Discretize age and RIN into 5 equal width bins

Must keep first degree relatives in order to model effects of collection site, processing site, and batch

Estimate number of latent confounders

Average BCV errors over 20 random seeds:



I chose $\widehat{K} = 20$.

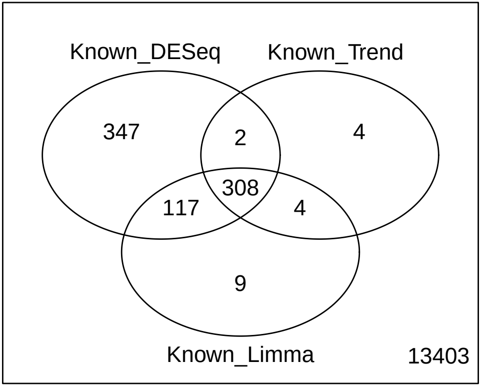
Counts of significant DE genes

```
##         test down  up
##  Known_DESeq  580 194
##  Known_Trend  178 141
##  Known_Limma  275 164
##   Known_Cate    0   0
##   Cate_DESeq  455 357
##   Cate_Trend  363 392
##   Cate_Limma  455 349
##         Cate   29 114
##    SVA_DESeq   24   8
##    SVA_Trend   19  11
##    SVA_Limma   18   5
```

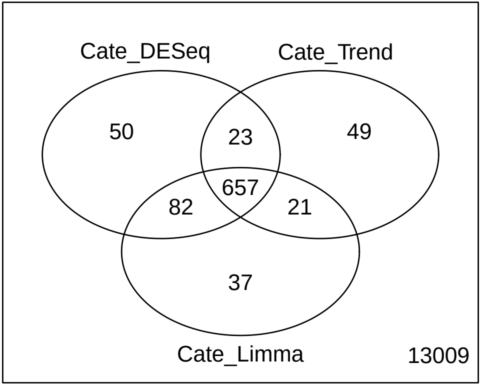
In test name:

- Part before underscore indicates what's in model:
  - only measured covariates,
  - cate confounders, or
  - sva confounders
- Part after underscore indicates DE testing method
- Cate by itself means cate used as intended

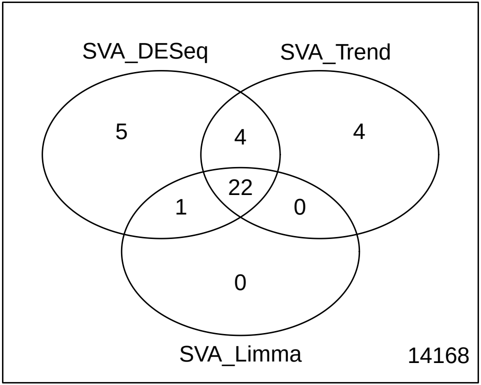
Comparison of discoveries when using only measured covariates



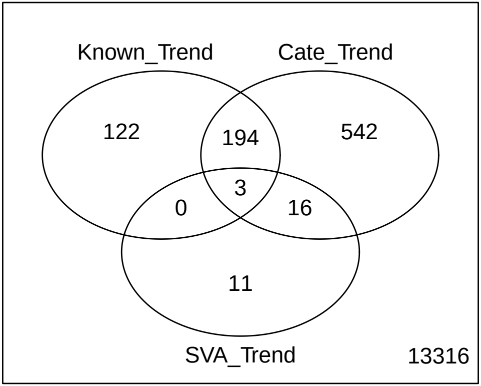
Comparison of discoveries when using cate confounders



Comparison of discoveries when using sva confounders



Comparison of discoveries when using limma with intensity-trend



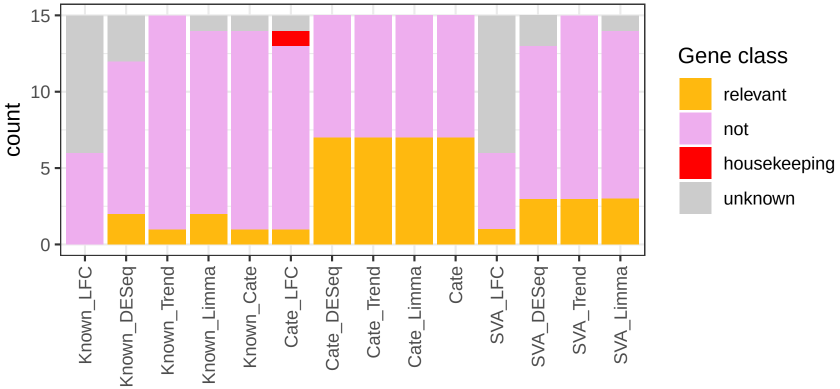
Checking highly ranked genes

Asked SME to determine biological relevance of top 15 genes from:

- limma using intensity-trend including:
    - only measured covariates,
    - cate confounders, or
    - sva confounders
- DESeq2 prioritizing by $|\text{LFC}|$

(Actually sent a few more genes than this)

Comparison of relevance of top 15 genes



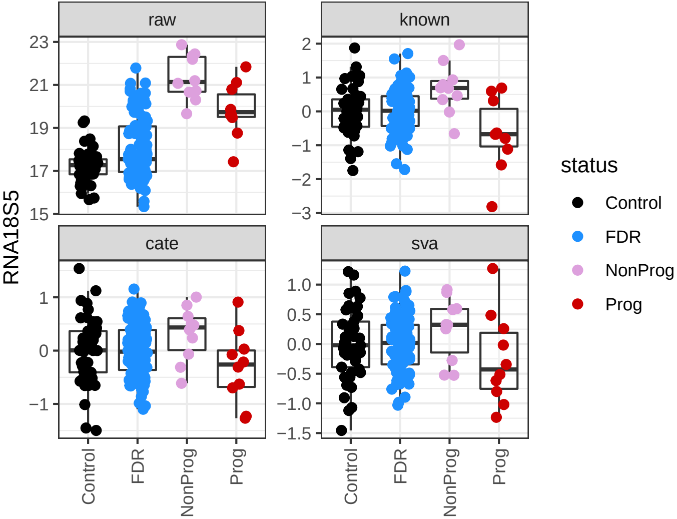
Comparison of relevance

Counts of relevant genes *checked by SME*

```
##         test top10 top15 top50
##    Known_LFC     0     0     0
##  Known_DESeq     1     2     5
##  Known_Trend     1     1     6
##  Known_Limma     1     2     5
##   Known_Cate     1     1     5
##     Cate_LFC     0     1     3
##   Cate_DESeq     6     7    10
##   Cate_Trend     7     7    11
##   Cate_Limma     5     7    10
##         Cate     5     7    10
##      SVA_LFC     1     1     2
##    SVA_DESeq     3     3     4
##    SVA_Trend     3     3     4
##    SVA_Limma     2     3     4
```

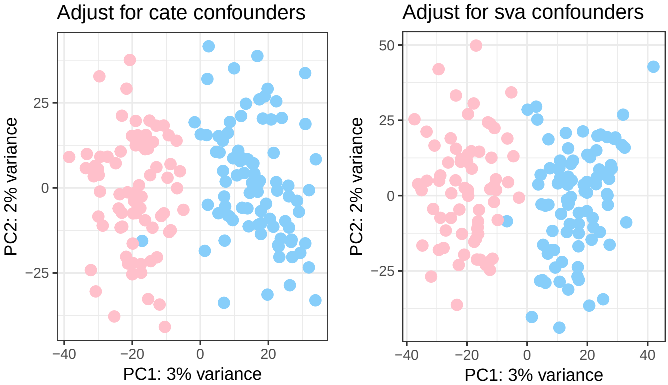
- Ranking by |LFC| is very different from other rankings
- Even in top 50, sva confounders still don't discover genes in cate's top 15
- Using only measured covariates eventually discovers some genes in cate's top 15 but also discovers more genes that are probably spurious

Housekeeping gene RNA18S5



Testing for DE Between Sexes

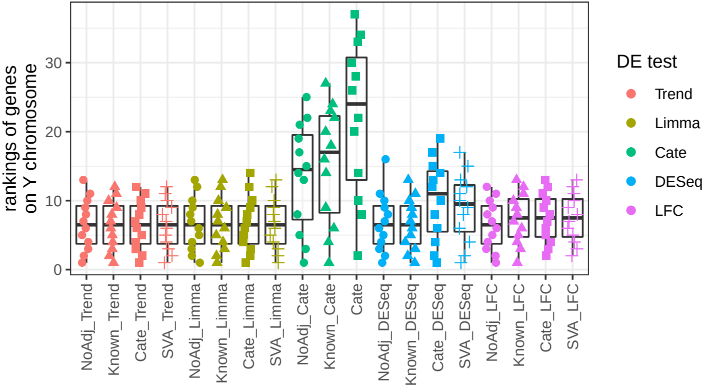
PCA plots



Pink for girls, blue for boys.

Discoveries on Y chromosome

12 genes on Y chromosome are discovered by any and every method



Other DE genes

- Probably not many autosomal genes DE between sexes
- X-chromosome genes that are DE between sexes should mostly be down-regulated in males

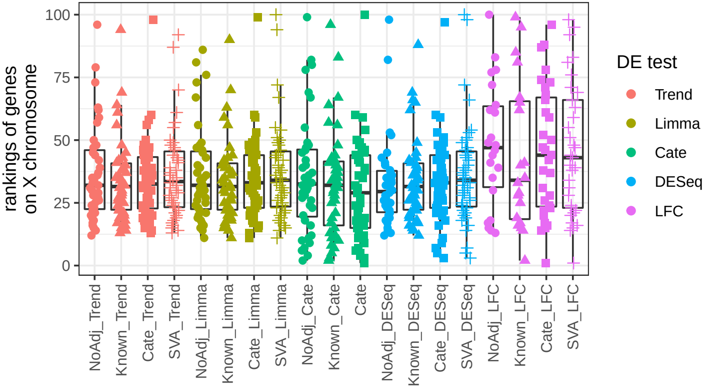
Are discoveries consistent with conjectures on preceding slide? (part 1)

```
##          test n.auto n.X.up n.X.down top100.X top100.X.up
##     NoAdj_LFC    457     14       42       22           2
##   NoAdj_DESeq    457     14       42       34           3
##   NoAdj_Trend    215      8       39       39           4
##   NoAdj_Limma    170      7       39       39           4
##    NoAdj_Cate      8      3       30       40           3
##     Known_LFC    884     18       54       20           1
##   Known_DESeq    884     18       54       38           4
##   Known_Trend    604     15       45       38           5
##   Known_Limma    658     15       50       38           4
##    Known_Cate      7      3       30       39           3
```

Are discoveries consistent with conjectures on preceding slide? (part 2)

```
##        test n.auto n.X.up n.X.down top100.X top100.X.up
##    Cate_LFC    983     25       63       31           4
##  Cate_DESeq    983     25       63       41           4
##  Cate_Trend    911     24       58       40           5
##  Cate_Limma    885     21       60       41           5
##        Cate     50      5       37       41           4
##     SVA_LFC    347     15       48       33           3
##   SVA_DESeq    347     15       48       43           5
##   SVA_Trend    301     14       45       42           5
##   SVA_Limma    305     14       47       43           5
```

Rankings of X-chromosome genes in top 100



Comparison of methods

- For given adjustment (or no adjustment), the X-chromosome genes discovered by **cate** are also discovered by all the other methods
- Other methods mostly agree on their discoveries
- Judging by the autosomal and up-regulated X-chromosome genes, **cate** confounders with robust regression seems to give far fewer false discoveries—and might even be overly conservative.
- Using **cate** confounders in **limma** or **DESeq2** versus using **sva** confounders in the same method:
    - Advantages of **cate** confounders:
        - About 30% more down-regulated X-chromosome discoveries
        - Slightly higher rankings of X-chromosome genes in top 100
    - Advantages of **sva** confounders seem to outweigh preceding list:
        - Roughly 1/3 as many autosomal discoveries
        - Roughly 2/3 as many X-chromosome discoveries up-regulated in males
        - 1–2 more down-regulated X-chromosome discoveries in top 100
- Ranking by |LFC| includes only 75% (or less) as many X-chromosome genes in top 100 and spreads them out more compared to other methods using the same model

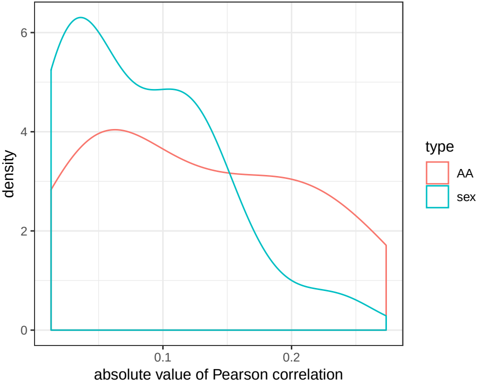
Wrap-up

Conclusions

- Ranking by $|\text{LFC}|$ does not work for these comparisons
  - Might be affected by using apeglm shrinkage method
  - I didn't try either of the other 2 shrinkage options
- AA+ vs controls: **cate** confounders clearly better than using **sva** confounders
- Between sexes:
  - **cate** with robust regression gives the fewest likely false discoveries and probably gives the best top 100 genes
  - Otherwise, **sva** confounders seems to give far fewer false discoveries than using **cate** confounders in the same method

Correlation between cate confounders and primary variable

Benchmarks with simulated data have shown that **sva** may perform poorly when latent factors are correlated with primary variable.



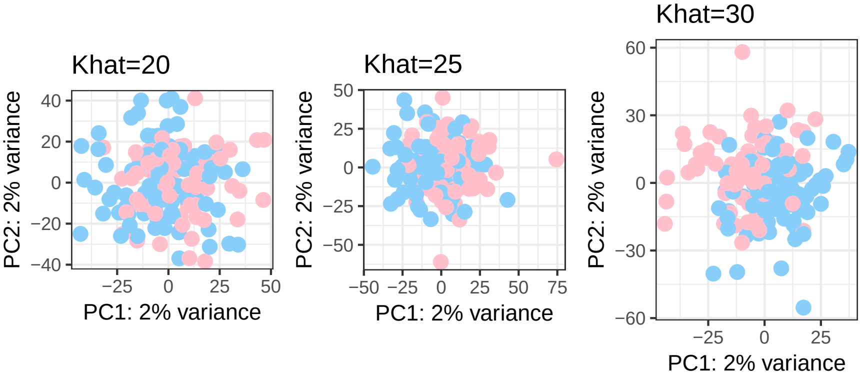
Confirms that **cate** confounders are more strongly correlated with being AA+ in first set of tests than with sex in second set.

Are latent confounders overfitting?

- Will using latent factors result in DE genes when none exist?
- Finding so many autosomal DE genes when comparing sexes might increase concern
- To check, randomly shuffled the sex labels and repeated DE testing between "fake" sexes
  - BCV indicates more latent factors but probably not more than 30
  - Tried **cate** with $\widehat{K} = 20, 25, 30$
  - PCA plots on next slide have a lot of overlap between groups
  - Some DE testing with 30 **cate** confounders:
    - **cate** discovered one gene (DDX3Y on Y chromosome)
    - **limma** (both versions) found none
- Details may depend on random seed before shuffling

PCA plots

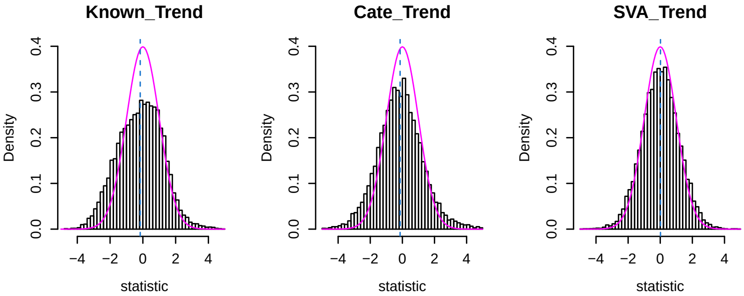
Randomly shuffle sex labels; use specified number of **cate** confounders



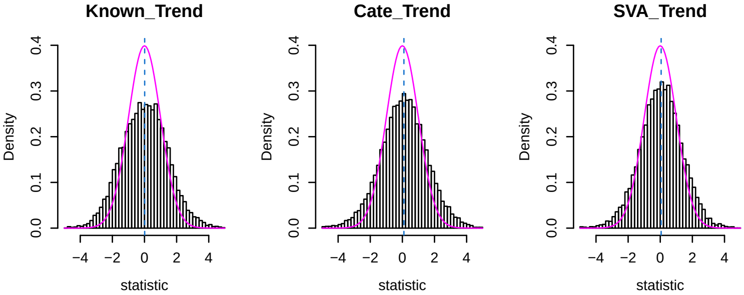
Distributions of test statistics

- I've been using PCA plots as diagnostic tool
- Should also check distributions of test statistics, ignoring "large" values
  - Does it match distribution assumed in computation of p-values?
  - With **cate** calibration (used by default and these results), roughly normal distribution almost guaranteed
  - Can alternatively check distribution of p-values
    - Histogram should be roughly uniform with spike at left end
    - I found it harder to judge these histograms than histogram of test statistics
- Following slides show a few examples
  - Ignoring statistics with absolute value greater than 5
  - Magenta curve is density function of standard normal
  - Blue dashed line is mean of histogram

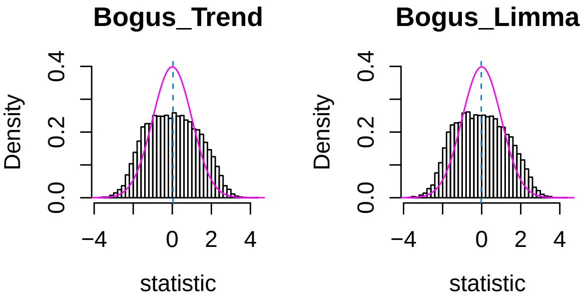
Statistics from testing DE between AA+ and controls



Statistics from testing DE between sexes



Statistics from DE testing with shuffled sex labels



Source and additional supporting material

- These slides came from http://web.stanford.edu/~lstell/
- Some R code demonstrating how to do these analyses to be posted
- Updates and additional materials might also be posted