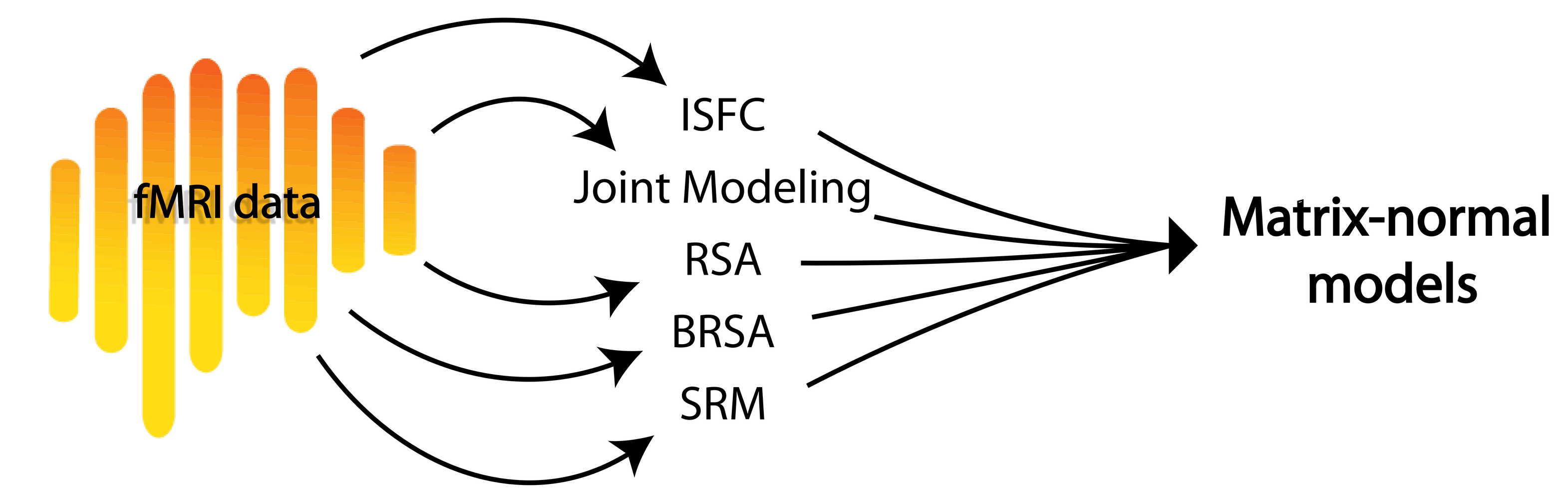
Matrix-normal models for fMRI analysis

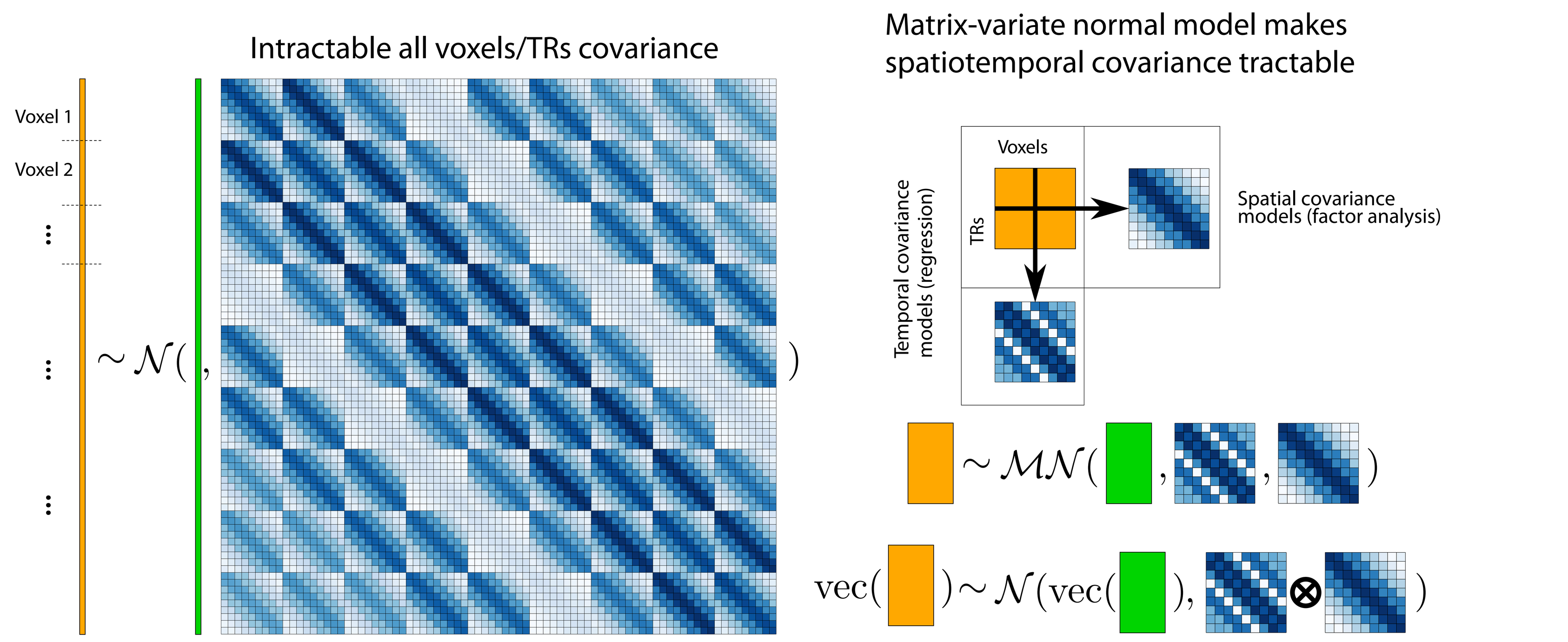
Michael Shvartsman[1], Narayanan Sundaram[2], Mikio C. Aoi[1], Adam Charles[1], Theodore L. Wilke[2], Jonathan D. Cohen[1]

Princeton University   Parallel Computing Lab, Intel Corporation

Our main contribution

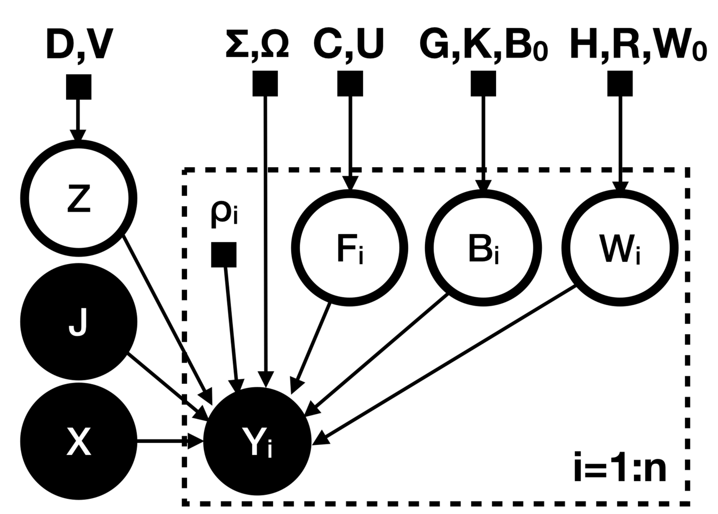


The matrix-variate normal distribution



One generative model includes many existing analyses

$\mathbf{Y}_i \mid \mathbf{F}_i, \mathbf{B}_i, \mathbf{W}_i, \Sigma_i, \Omega \sim$
$\mathcal{MN}(\mathbf{F}_i\mathbf{Z} + \mathbf{B}_i\mathbf{X} + \mathbf{J}\mathbf{W}, \rho_i^2\Sigma, \Omega)$
$\mathbf{F}_i \mid \mathbf{C}, \mathbf{U} \sim \mathcal{MN}(0, \mathbf{C}, \mathbf{U})$
$\mathbf{Z} \mid \mathbf{D}, \mathbf{V} \sim \mathcal{MN}(0, \mathbf{D}, \mathbf{V})$
$\mathbf{B}_i \mid \mathbf{G}, \mathbf{K} \sim \mathcal{MN}(\beta_0, \mathbf{G}, \mathbf{K})$
$\mathbf{W}_i \mid \mathbf{H}, \mathbf{R} \sim \mathcal{MN}(\mathbf{W}_0, \mathbf{H}, \mathbf{R})$.



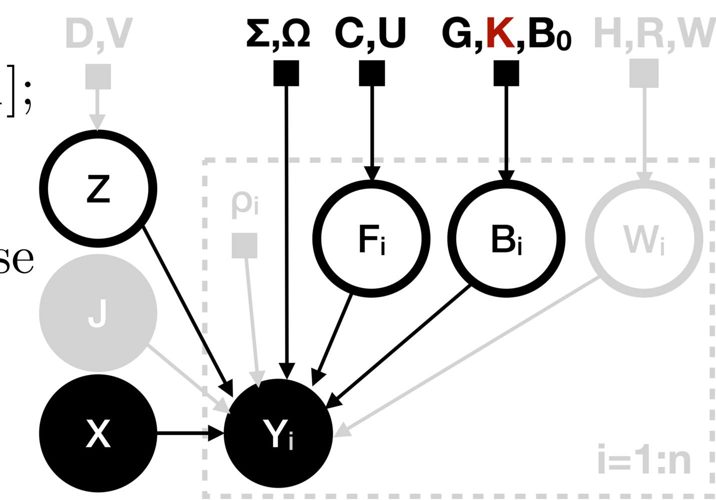
- $\mathbf{Y}_i$: data for subject i. $\mathbf{X}/\mathbf{J}$ are temporal/spatial design matrices.
- This poster: temporal cov. Ω is AR(1), spatial Σ is diagonal.

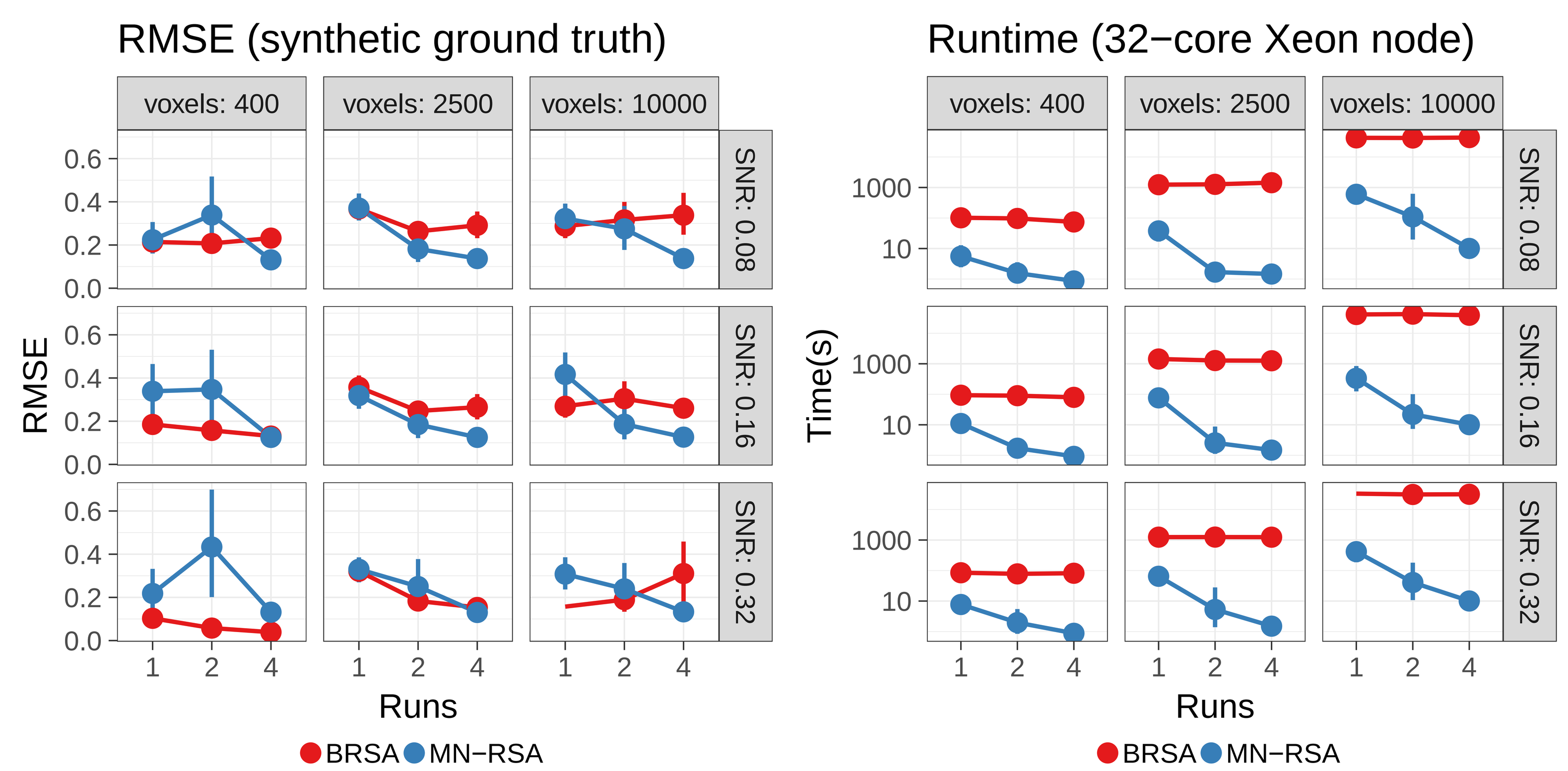
BrainIAK

Analysis performed using the BrainIAK python package for high-performance neuroimaging analysis. For additional information, see `brainiak.org/sfn2017`

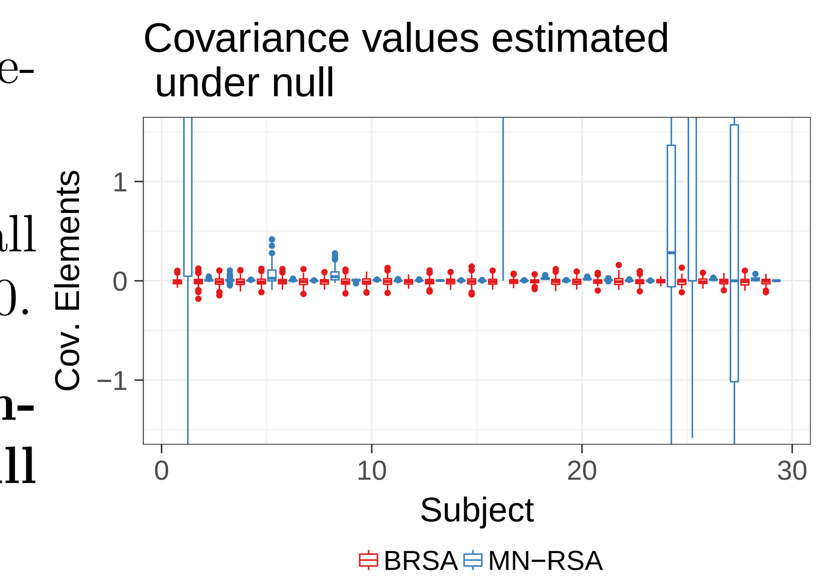
Matrix-Normal RSA: faster and more accurate at large data and/or low SNR

- **Mitigates bias like BRSA** ([1]; poster 260.05).
- **Fewer parameters** (different noise model).
- **Try both!**

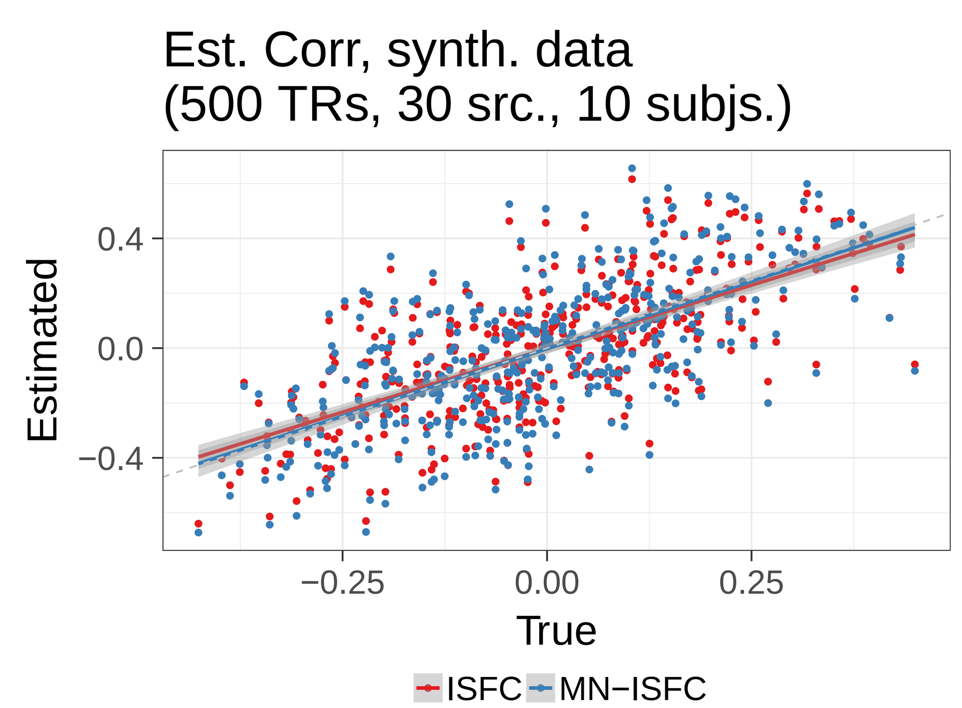


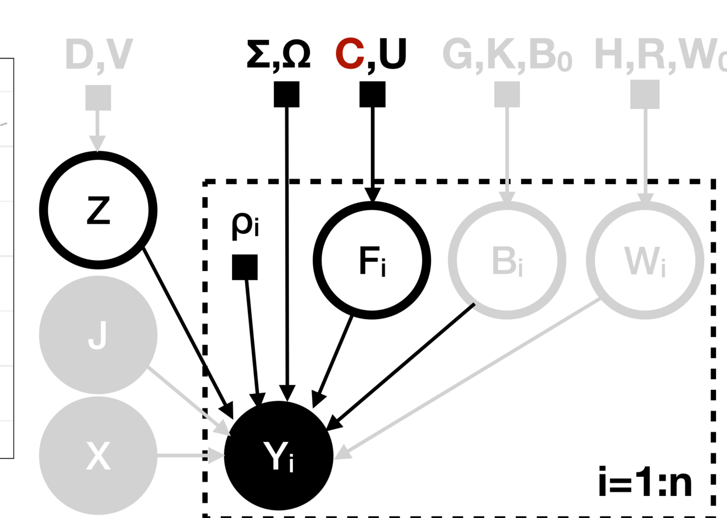


- Resting state data, unrelated design matrix
- **FZ** should absorb all variance, $\mathbf{B} \to 0, \mathbf{K} \to 0$.
- **MN-RSA more conservative under null** for most subjects.



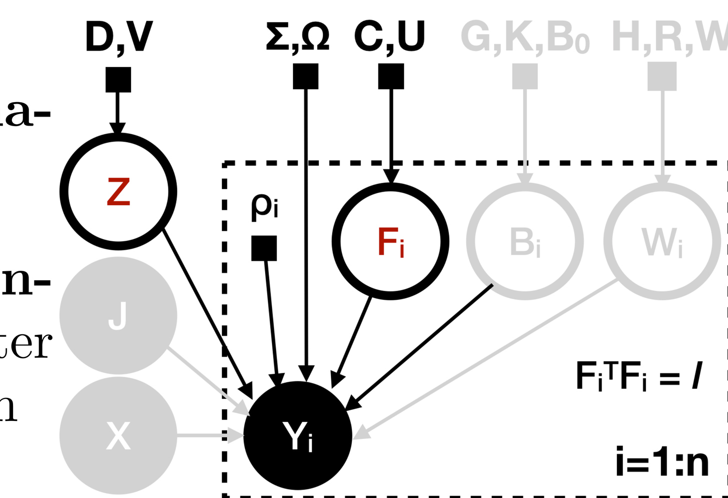
Matrix-normal ISFC: maximum-likelihood estimation, valid correlations

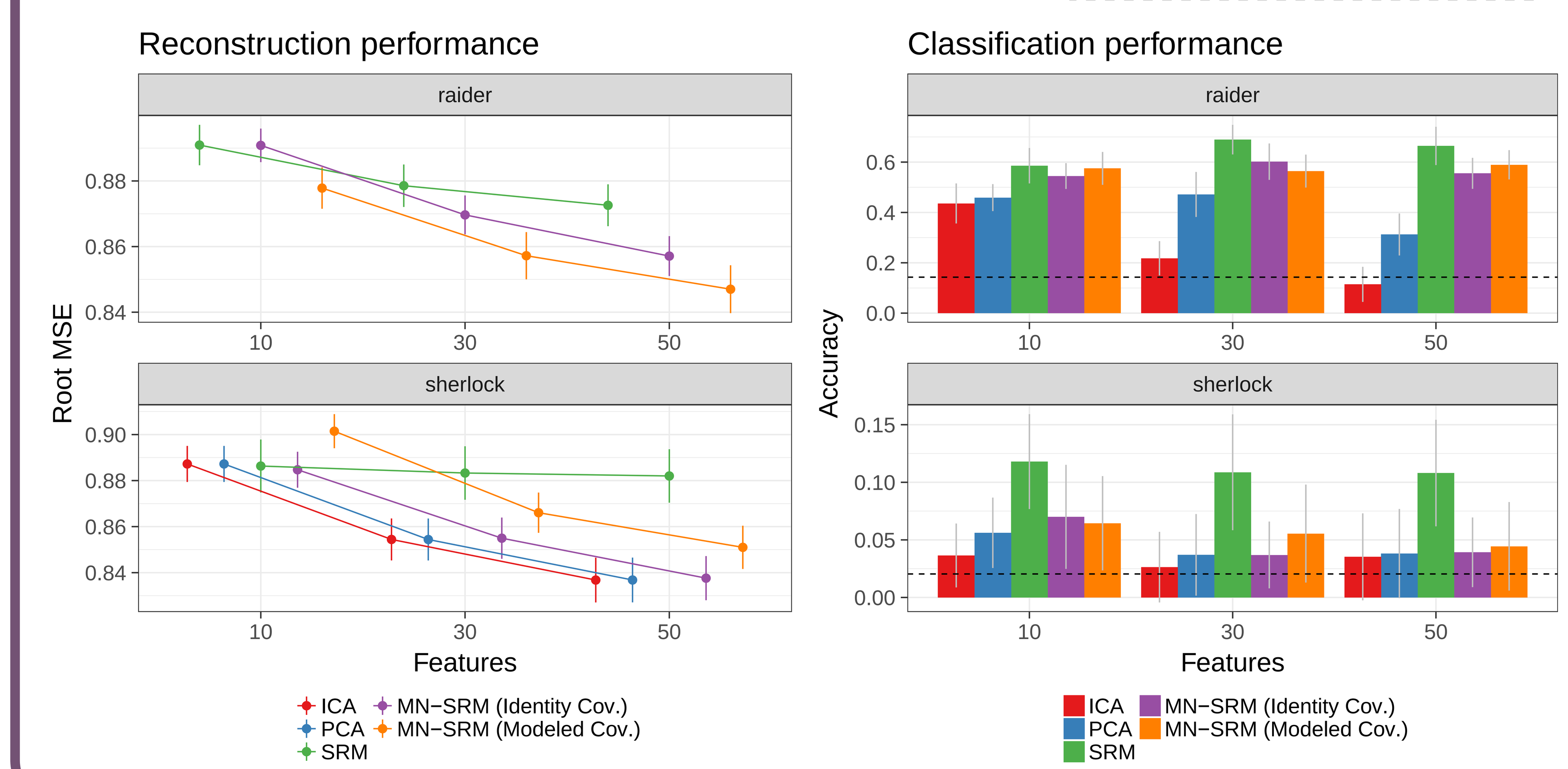




Matrix-Normal SRM: improved reconstruction, fewer parameters

- **ECM algorithm for fast estimation.**
- **Better out-of-sample reconstruction than SRM** ([2]; poster 260.08) but worse feature selection





brainiak.matnormal: a prototyping tool for matrix-normal models

MN-RSA can be implemented in ≈50 lines of code!

```
rsa_cov = CovFullRankCholesky(size=k)
space_noise_cov = CovDiagonal(size=v)
time_noise_cov = CovAR1(size=t)
params = [rsa_cov.get_optimize_vars(),
          time_noise_cov.get_optimize_vars(),
          space_noise_cov.get_optimize_vars()]
loss = -(time_noise_cov.logp +
         space_noise_cov.logp +
         rsa_cov.logp +
         matnorm_logp_marginal_row(Y, row_cov=time_noise_cov,
                                   col_cov=space_noise_cov,
                                   marg=X, marg_cov=rsa_cov))
optimizer.minimize(loss)
U = rsa_cov.Sigma
C = cov2corr(U)
```

Automatic marginalization and covariance structure selection.

References

Cai, M. B., Schuck, N. W., Pillow, J. W., & Niv, Y. NIPS 2016; [2] Chen, P.-H., Chen, J., Yeshurun, Y., Hasson, U., Haxby, J., & Ramadge, P. J. NIPS 2015; [3] Simony, E., Honey, C. J., Chen, J., Lositsky, O., Yeshurun, Y., Wiesel, A., & Hasson, U. Nat. Comms. 7:12141 (2016).