Supplement to "Improving power while controlling the false discovery rate when only a subset of peptides are relevant"

Andy Lin[1], Deanna L. Plubell[1], Uri Keich[2], and William S. Noble*[1,3]

[1]Department of Genome Sciences, University of Washington, Seattle, WA, USA
[2]School of Mathematics and Statistics, University of Sydney, NSW, Australia
[3]Paul G. Allen School for Computer Science and Engineering, University of Washington, Seattle, WA, USA

*Corresponding author: Box 355065, Foege Building, S220B, 3720 15th Ave NE, Seattle, WA 98195-5065, william-noble@uw.edu

Supporting Information

- **Supplemental File S1:** PDF containing pseudocode for the five different search procedures, plus four supplemental tables.
- **Supplemental File S2:** Python script (filterPSMsByGroup.py) used to select PSMs for subset search, group FDR, all-sub, and SNS for input into target-decoy compeition and subsequent FDR estimation.
- **Supplemental File S3:** Python script (allSub.py) used to estimate all-sub q-values.

Algorithm 1 Search-then-select

procedure searchThenSelect($S, \mathcal{T}, \mathcal{D}, \mathcal{T}_r, \alpha$)
  $(M, P) :=$ Search($S, \mathcal{T} \cup \mathcal{D}$)
  $A :=$ ControlFDR(M, P, α)
  $R := \{(s_i, p_i, m_i) \mid a_i = 1, p_i \in \mathcal{T}_r\}$
  return R
end procedure

Algorithm 2 Subset-search

procedure subsetSearch($S, \mathcal{T}_r, \mathcal{D}_r, \alpha$)
  $(M, P) :=$ Search($S, \mathcal{T}_r \cup \mathcal{D}_r$)
  $A :=$ ControlFDR(M, α)
  $R := \{(s_i, p_i, m_i) \mid a_i = 1\}$
  return R
end procedure

Algorithm 3 Group-FDR

procedure groupFDR ($S, \mathcal{T}_r, \mathcal{D}_r, \mathcal{T}_i, \mathcal{D}_i, \alpha$)
  $(M, P) :=$ Search($S, \mathcal{T}_r \cup \mathcal{T}_i \cup \mathcal{D}_r \cup \mathcal{D}_i$)
  $(M^1, P^1) := \{(m_i, p_i) \mid p_i \in \mathcal{T}_r \cup \mathcal{D}_r\}$
  $A :=$ ControlFDR(M^1, α)
  $R := \{(s_i^1, p_i^1, m_i^1) \mid a_i = 1\}$
  return R
end procedure

Algorithm 4 All-sub

procedure allSub($S, \mathcal{T}, \mathcal{D}, \mathcal{T}_r, \alpha$)
  $(M, P) :=$ Search($S, \mathcal{T} \cup \mathcal{D}$)
  $A :=$ ControlFDR2(M, α)
  $R := \{(s_i, p_i, m_i) \mid a_i = 1\}$
  return R
end procedure

Algorithm 5 Subset-neighbor search (SNS)

procedure SNS ($S, \mathcal{T}_r, \mathcal{D}_r, \mathcal{T}_n, \mathcal{D}_n, \alpha$)
  $(M, P) :=$ Search($S, \mathcal{T}_r \cup \mathcal{T}_n \cup \mathcal{D}_r \cup \mathcal{D}_n$)
  $(M^1, P^1) := \{(m_i, p_i) \mid p_i \in \mathcal{T}_r \cup \mathcal{D}_r\}$
  $A :=$ ControlFDR(M^1, α)
  $R := \{(s_i^1, p_i^1, m_i^1) \mid a_i = 1\}$
  return R
end procedure

file name	# scans
UPS1_12500amol_R1.ms2	39718
UPS1_12500amol_R2.ms2	39682
UPS1_12500amol_R3.ms2	39782
UPS1_25000amol_R1.ms2	40856
UPS1_25000amol_R2.ms2	40512
UPS1_25000amol_R3.ms2	40439
UPS1_50000amol_R1.ms2	41833
UPS1_50000amol_R2.ms2	41653
UPS1_50000amol_R3.ms2	41665
UPS1_5000amol_R1.ms2	37918
UPS1_5000amol_R2.ms2	38046
UPS1_5000amol_R3.ms2	38052

Table S1: **yeast/UPS1 data.** The number of scans found in each run. This is for the samples where UPS1 was spiked into yeast.

*file name	cultivar	preparation method	# scans	*file name	cultivar	preparation method	# scans
Zanz_1_1_03Jun16	*Zanzibarensis*	M0	15735	GCH4_1_1_03Jun16	GCH4	M0	21634
Zanz_1_2_03Jun16	*Zanzibarensis*	M0	17138	GCH4_1_2_03Jun16	GCH4	M0	16765
Zanz_1_3_03Jun16	*Zanzibarensis*	M0	18834	GCH4_1_3_03Jun16	GCH4	M0	18663
Zanz_2_1_03Jun16	*Zanzibarensis*	M0	19760	GCH4_2_1_03Jun16	GCH4	M0	19830
Zanz_2_2_03Jun16	*Zanzibarensis*	M0	18986	GCH4_2_2_03Jun16	GCH4	M0	24944
Zanz_2_3_03Jun16	*Zanzibarensis*	M0	16222	GCH4_2_3_03Jun16	GCH4	M0	21667
Zanz_3_1_03Jun16	*Zanzibarensis*	M0	18900	GCH4_3_1_03Jun16	GCH4	M0	20267
Zanz_3_2_03Jun16	*Zanzibarensis*	M0	21368	GCH4_3_2_03Jun16	GCH4	M0	22246
Zanz_3_3_03Jun16	*Zanzibarensis*	M0	20315	GCH4_3_3_03Jun16	GCH4	M0	25330
TMVCH1_1_1_03Jun16	TMVCH1	M0	14893	200_1_03Jun16	200	M0	13742
TMVCH1_1_2_03Jun16	TMVCH1	M0	20086	200_2_03Jun16	200	M0	17822
TMVCH1_1_3_03Jun16	TMVCH1	M0	23470	200_3_03Jun16	200	M0	6158
TMVCH1_2_1_03Jun16	TMVCH1	M0	19125	5952_1_03Jun16	592	M0	14951
TMVCH1_2_2_03Jun16	TMVCH1	M0	23804	592_2_03Jun16	592	M0	16157
TMVCH1_2_3_03Jun16	TMVCH1	M0	20689	592_3_03Jun16	592	M0	14125
TMVCH1_3_1_03Jun16	TMVCH1	M0	16456	611_1_03Jun16	611	M0	10724
TMVCH1_3_2_03Jun16	TMVCH1	M0	25938	611_2__03Jun16	611	M0	13621
TMVCH1_3_3_03Jun16	TMVCH1	M0	16414	611_3_03Jun16	611	M0	8454
1_M0_AM_R1_7Mar16	PNNL	M0	5076	829_1_03Jun16	829	M0	17343
2_M2_JA_R3_7Mar16	PNNL	M2	16784	3_M0_JA_R3_7Mar16	829	M0	18592
3_M0_JA_R3_7Mar16	PNNL	M0	15859	829_3_03Jun16	829	M0	9475
3_M0_JA_R3_B_03Jun16	PNNL	M0	22015	8_M1_JA_R2_7Mar16	PNNL	M1	15669
4_M1_AM_R1_7Mar16	PNNL	M1	13188	9_M4_AM_R1_7Mar16	PNNL	M4	14892
5_M2_AM_R1_7Mar16	PNNL	M2	15358	9_M4_JA_R2_7Mar16	PNNL	M4	16140
6_M0_JA_R2_7Mar16	PNNL	M0	16728	9_M4_JA_R3_7Mar16	PNNL	M4	15472
6_M0_JA_R2_B_03Jun16	PNNL	M0	21544	10_M2_JA_R2_7Mar16	PNNL	M2	18358
7_M1_JA_R3_7Mar16	PNNL	M1	17973	16_M3_03Jun16	PNNL	M3	20583

Table S2: **Ricin data.** The number of scans found in each ricin run. *All file names start with "Rcom_" and end with either "_Samwise_16-03-32.ms2" or "_Samwise_15-08-55.ms2".

file name	# scans
217_2018_ZBS6_HeLa_ISD_1.ms2	89162
220_2018_ZBS6_HeLa_SPEED_1.ms2	87981
222_2018_ZBS6_HeLa_FASP_2.ms2	87398
223_2018_ZBS6_HeLa_ISD_2.ms2	86869
225_2018_ZBS6_HeLa_SP3_2.ms2	90238
226_2018_ZBS6_HeLa_SPEED_2.ms2	88086
228_2018_ZBS6_HeLa_FASP_3.ms2	87255
229_2018_ZBS6_HeLa_ISD_3.ms2	87835
231_2018_ZBS6_HeLa_SP3_3.ms2	90546
232_2018_ZBS6_HeLa_SPEED_3.ms2	87703
234_2018_ZBS6_HeLa_FASP_1.ms2	87797
235_2018_ZBS6_HeLa_SP3_1.ms2	89253

Table S3: **Human data.** The number of scans found in each run.

file name	# scans
OR20070924_S_mix7_02.ms2	4966
OR20070924_S_mix7_03.ms2	5109
OR20070924_S_mix7_04.ms2	5094
OR20070924_S_mix7_05.ms2	5148
OR20070924_S_mix7_06.ms2	5170
OR20070924_S_mix7_07.ms2	5164
OR20070924_S_mix7_08.ms2	5268
OR20070924_S_mix7_09.ms2	5159
OR20070924_S_mix7_10.ms2	5257
OR20070924_S_mix7_11.ms2	957

Table S4: **ISB18 data.** The number of scans found in each run.