

1 **PeptideMind – applying machine learning algorithms to assess replicate quality in shotgun**  
2 **proteomic data**

3

4 **Handler, D.C.L., Haynes\*, P.A.**

5 Department of Molecular Sciences, Macquarie University, Sydney, NSW 2109, Australia

6 \*To whom correspondence should be addressed

7 Professor Paul A. Haynes

8 Department of Molecular Sciences

9 Macquarie University

10 North Ryde, NSW 2109, Australia

11 Email: paul.haynes@mq.edu.au

12 Phone: 61-2-9850 6258

13 Fax: 61-2-9850 6200

14

15

16 **Abstract**

17 Assessment of replicate quality is an important process for any shotgun proteomics experiment.  
18 One fundamental question in proteomics data analysis is whether any specific replicates in a set  
19 of analyses are biasing the downstream comparative quantitation. In this paper, we present an  
20 experimental method to address such a concern. PeptideMind uses a series of clustering Machine  
21 Learning algorithms to assess outliers when comparing proteomics data from two states with six  
22 replicates each. The program is a JVM native application written in the Kotlin language with  
23 Python sub-process calls to scikit-learn. By permuting the six data replicates provided into four  
24 hundred triplet non redundant pairwise comparisons, PeptideMind determines if any one  
25 replicate is biasing the downstream quantitation of the states. In addition, PeptideMind  
26 generates useful visual representations of the spread of the significance measures, allowing  
27 researchers a rapid, effective way to monitor the quality of those identified proteins found to be  
28 differentially expressed between sample states.

29 **Keywords:**

30 Classification, data quality, data validation, false discovery, kotlin, label-free shotgun proteomics,  
31 machine learning, protein quantitation, spectral counting, statistics.

32

33

34 **Required Metadata**

35

36 **Current code version**

37

38 *Table 1 – Code metadata (mandatory)*

<b>Nr</b>	<b>Code metadata description</b>	<b><i>Please fill in this column</i></b>
C1	Current code version	<i>V1.0.1</i>
C2	Permanent link to code/repository used of this code version	<i>www.bitbucket.org/peptidewitch/peptidemind</i>
C3	Code Ocean compute capsule	<i>N/A</i>
C4	Legal Code License	<i>MIT</i>
C5	Code versioning system used	<i>git</i>
C6	Software code languages, tools, and services used	<i>Kotlin, Gradle, Python</i>
C7	Compilation requirements, operating environments & dependencies	<i>Gradle, python packages listed in requirements.txt</i>
C8	If available Link to developer documentation/manual	<i>www.bitbucket.org/peptidewitch/peptidemind</i>
C9	Support email for questions	<i>david.handler@students.mq.edu.au</i>

39

40 Note - Sample data used to generate the figures in this manuscript can be downloaded from

41 <https://bitbucket.org/peptidewitch/peptidemind/downloads>

## 42 **1. Motivation and Significance**

43 Proteomics is the large-scale study of expressed proteins in biological systems, where the  
44 researcher undertakes analysis of biological networks that underpin cellular processes. In  
45 addition, there is a large data science component whereby raw data from a mass spectrometer  
46 has to be matched against known protein and peptide sequences before downstream analysis  
47 can occur. Making sense of all the requirements of a proteomics experiment can be a challenge;  
48 one such challenge is ensuring the validity of biological information drawn from quantitative  
49 comparisons of protein expression between test states. For example, if we have a cancer cell line  
50 dosed with a new synthetic drug, how can we effectively say that protein expression profiles  
51 generated from our treated cell state differ from those of our control cell state? There are several  
52 factors to consider, including experimental design, the process of identification of proteins, and  
53 the quantitation of proteins between these two states. Much attention has been paid to  
54 downstream forms of biological validation such as Western blotting (using an antibody to show  
55 qualitative differences between specific proteins) and parallel reaction monitoring (an  
56 orthogonal targetted mass spectrometry based method) as means to confirm an increase or  
57 decrease in protein expression between states [1]. However, there is also much to be gained  
58 from applying a more rigorous set of statistical or analytical tools prior to laboratory follow-up  
59 experiments. As Kall and colleagues describe in their seminal paper on the Percolator software  
60 [2], the application of smart methods to discriminate good quality from poor quality data can  
61 greatly improve the quality of the dataset, which can clarify any downstream quantitation  
62 performed using the data in question.

63 Over the past two decades, researchers have leveraged the power of machine learning  
64 algorithms (MLAs) in order to assist with shotgun proteomics data analysis. Classification  
65 algorithms are a natural fit for determining biomarker identity from state comparison  
66 experiments [3], while support vector machines have been utilised in a semi-trained fashion to  
67 help determine false discoveries in peptide to spectrum matching [2]. Machine learning  
68 algorithms have also been used to aid with peptide to spectrum matching, for example in the  
69 application of decision trees to utilise ion fragmentation patterns for peptide identification [4].

70 Recently, with the advent of easier implementation of MLAs via python packages such as  
71 Python's scikit-learn, more proteomics researchers are choosing to enhance their data analysis  
72 with MLAs [5].

73 When it comes to their application, MLAs are by no means a silver bullet that allows a researcher  
74 to exclusively target critical insights. Not every problem can be solved by running data through  
75 Support Vector Machines or Generalised Linear Models, and not every dataset is suited to, for  
76 example, Random Forest Classification. Sometimes a simple Decision Tree can suffice. There is  
77 also the consideration of training sample sizes – the oft-quoted adage of 'the more data, the  
78 better' makes no *a priori* assertion regarding the quality of the data, and it may well be the case  
79 that smaller, more accurate data pools are better to train against [6]. Shotgun proteomics  
80 datasets, then, are an interesting dataset to work with, given their relatively small size compared  
81 with transcriptomics and genomics, as well as their proportion of valuable biological information  
82 relative to their noise level, or junk data. Unlike binary sentiment/natural language analysis (like  
83 understanding if a movie review is positive or negative by word choice alone), or qualitative  
84 categorisation (such as movie ratings), proteomics data is a mixture of noise and overlapping  
85 signal. A deep learning approach to six replicates of two states may not provide a clear insight  
86 into the relationship between these two states, whereas a more simple classifier-based approach  
87 could potentially offer insights as to the quality of the comparisons being made.

88 In this report, we introduce PeptideMind, a Kotlin/Python hybrid program that utilises a cohort  
89 of classifier MLAs to analyse replicate quality from proteomic peptide to spectrum matching  
90 search engine outputs. Following on from the same-same architecture that our lab group has  
91 developed [7], PeptideMind will take six replicates of control and treatment protein ID data and  
92 conduct multiplexed non-redundant analysis on the protein identifications (IDs). The output of  
93 the PeptideMind process is a series of graphics by which the researcher can assess overall  
94 replicate quality and check for outliers, as well as focus on specific protein identifications of  
95 interest and analyse the spread of difference in expression levels, thus assisting the researcher  
96 in providing a confident measure of statistical validation.

## 97 **2. Software Description**

98 PeptideMind has two components. The first is a Kotlin/Java environment that can be replicated  
99 through use of the Gradle file included within the repository. Working with a development  
100 environment such as IntelliJ by JetBrains should handle the installation automatically; manual  
101 installation can be achieved with other environments. Regardless, Gradle is mandatory. Secondly,  
102 PeptideMind requires a version of python to be installed on the user system. Currently,  
103 PeptideMind defaults to a specific version of Python installed on the Path (a future update will  
104 allow users to point to a virtualenv such as pipenv). This python environment should have the  
105 packages installed from the requirements.txt file in the PeptideMind source directory.

106 The software is comprised of a single GUI page made with TornadoFX, as shown in Figure 1. Users  
107 follow the control flow from left to right hand side of the page, selecting the following elements:

- 108 1. The folder location for the control state data
- 109 2. The folder location for the treatment state data
- 110 3. The type of peptide to spectrum matching search engine used
- 111 4. The type of MLAs to use for analysis (at least 1 must be selected)
- 112 5. The scope by which the analysis is conducted on the whole dataset
- 113 6. Whether any proteins of interest should be targetted. If users input a specific text  
114 identifier (in the same format as their PSM engine from step 3) the resulting analysis  
115 will only focus on these identifiers to the exclusion of all others.
- 116 7. A folder location for the output data, and
- 117 8. A 'start' button

118 The resulting output will be contained within the folder specified in Step 7. Users must separate  
119 their control and treatment states data into separate folders and name their replicates according  
120 to the following structure: *%name-R%replicate\_number*. For example, statin-R1, statin-R2, etc,  
121 for the control state. Both states must have exactly six replicates each.

122 PeptideMind outputs four broad categories of results.

- 123 1. A series of excel files containing the common protein identifiers found across all six  
124 replicates for both states, including their Student's T-Test significance values at both the  
125 spectral counting level and the exponential logarithmic normalised spectral abundance  
126 factor (NSAF) level [8,9].
- 127 2. Isolation Forests for each common ID displaying the aforementioned significance  
128 measures along the X and Y axis.
- 129 3. A MultiLabel deviation plot, where each replicate is shown against a middle line value of  
130 0.5. This plot is designed to give users, at a glance, some idea of which replicates in which  
131 state are contributing to up- or down-regulation of significant proteins.
- 132 4. A 'Mega Isolation' Forest comprising an aggregate spread of protein significance  
133 measures along all four hundred combinations.

134 A more in-depth explanation of the processes of PeptideMind and how it arrives at the end  
135 results is provided below.

#### 136 *Result category 1: .csv outputs*

137 PeptideMind begins with two sets of sextuplicate results from proteomics peptide to spectrum  
138 matching search engines, currently including Proteome Discoverer [10], Meta Morpheus [11] and  
139 X!Tandem [12]. Broadly speaking, what the PeptideMind program aims to do is to produce an  
140 internal measure of inter-replicate variability, and display the spread of protein expression level  
141 variance, for the user to determine if subsequent differential analysis is worthwhile. To achieve  
142 this aim, the program begins by sorting the six replicates of the control and six replicates of the  
143 treatment state into sub-experiments for analysis, as shown in Figure 2. A pair of states with six  
144 replicates apiece can be split into two sets of three in four hundred non-redundant pairs. These  
145 pairs of three by three comparisons each constitute their own analysis. This pairwise combination  
146 undergoes a round of data filtering by Minimum Spectral Counting [13,14] before being subjected  
147 to two different types of Student T-Tests – one based on the spectral count of each protein, and  
148 the other based on natural log NSAF values. Any protein identifier that is considered significantly  
149 differentially expressed is recorded. This analysis is then repeated by shuffling new combinations

150 of paired triplets between state one and state. Next, a list of proteins common to all four hundred  
151 triplet comparisons is produced, and the significance value from the two types of T-Tests are  
152 matched to the protein identifier. We then consider this data our training set for the machine  
153 learning algorithms. Data from this process is stored in .csv outputs for the user to examine for  
154 their own interest.

155 *Result category 2: isolation forests for common protein identifiers between the states*

156 Next, each of the common IDs from the four hundred tests is subjected to an individual Isolation  
157 Forest algorithm. The results from students T-Tests on InNSAF and spectral count data are plotted  
158 in two dimensions, with the background color-gradient coded to correspond to the spread of  
159 significance results within the data as determined by an Isolation Forest algorithm. The results  
160 from all four hundred test combinations are shown as black dots, while a single yellow dot  
161 corresponds to the significance measure of the protein from the original 6 by 6 replicate test  
162 conducted. This serves as an anchor for the results: most researchers would only see this single  
163 measure of significance, but now, graphically, we have a way of determining the spread of  
164 significance measures in a manner that is intuitive and visually informative.

165 *Result category 3: multi-MLA analysis of replicate contribution to significance measures.*

166 Another useful measure of inter-replicate variability comes from a blind assignment of protein  
167 identifiers with regards to their corresponding differentiation levels from replicate values as  
168 determined by four separate multi-classifier algorithms selected by the user in Step 4 of the  
169 workflow described earlier. The results are displayed as a histogram plot which indicates the  
170 relative contribution of each replicate towards the quantitative differences of all protein  
171 identifiers shared between the states. Each data point is the average value found from the  
172 selected MLAs. What the machine learning consensus network here achieves is a blind  
173 assignment of protein identifiers to replicates, thus allowing the researcher to see if any one  
174 replicate in particular is consistently contributing more weight to differential analysis.



### 175 **3. Illustrative Examples**

176 To provide illustrative examples, we analysed data from two ongoing studies in our laboratory.  
177 The first study involved proteomic analysis of *Eucalyptus grandis* leaf tissue, with six replicates  
178 corresponding to young healthy leaf tissue set and six replicates from old senescent leaf tissue.  
179 Proteomic data was acquired using the X!Tandem algorithm for peptide to spectrum matching.  
180 The second study involved proteomic analysis of two different laboratory yeast strains  
181 designated CCC and CCB, with proteomic data acquired using the Meta Morpheus algorithm for  
182 peptide to spectrum matching.

183 The individual Isolation Forests for data generated by the program for each shared protein  
184 identifier are shown in Figure 3 for two selected proteins from the Eucalyptus experiment: K1C9  
185 is a human keratin protein present at variable levels as a result of sample handling contamination;  
186 XP\_010027978.1 is a serine hydroxymethyltransferase metabolic protein. In Figure 3a the protein  
187 ID significance shows a wide spread of results for the K1C9 protein across the different  
188 comparisons, while Figure 3b shows a very tightly correlated spread of protein ID significance for  
189 the metabolic protein. This can be viewed as a form of statistical validation; if a protein ID of  
190 interest displays a spread of significance similar to Figure 3b, rather than Figure 3a, then we can  
191 say with confidence that this result is more likely to be significant and less influenced by inter-  
192 replicate noise.

193 Figure 4 shows the histogram plots of the average output values found from the selected MLAs,  
194 for both the Eucalyptus data set and the yeast data set. In Figure 4a, for the Eucalyptus data,  
195 there is clearly significant variation between the replicates. In the data from young leaf tissue  
196 (replicates R1-R6), replicates 1 and 4 contribute relatively less to the differentially regulated  
197 protein identifiers, whereas replicates 2 and 6 are overrepresented. In the data from old leaf  
198 tissue (replicates R7 to R12), replicate 8 contributes more weight to protein differentiation than  
199 replicates 10 and 12. Replicate 4, in particular, may not be truly representative of the proteome  
200 state given the relative disparity observed. In contrast, the histogram shown in Figure 4b for the  
201 yeast data indicates that all 12 replicates are internally consistent, and all contributed relatively  
202 equally to the observed differences in protein expression.

203 The red dotted line in the histograms represents an ideal result – if we are to assume that all of  
204 the replicates hold equal analysis weight in the course of the experiment, then every replicate  
205 should fall on the dotted line and report the same result for every one of the 400 triplet  
206 combinations. In reality this is not the case, as some replicates have higher numbers of specific  
207 proteins relative to their state cohort. This chart is a visual indicator of how biased our results  
208 are, in terms of which replicates are over- or under-represented in their contribution to protein  
209 significance spreads. A more even histogram indicates high quality replicates and low levels of  
210 variable noise, whereas a histogram more reminiscent of a city skyline may indicate significant  
211 problems in the reproducibility of the data.

#### 212 **4. Impact**

213  
214 PeptideMind provides user with clear visual metrics concerning the validity of their downstream  
215 quantitation profiles, by highlighting the spread of variance for every protein identifier and every  
216 replicate within the total system. At a glance, the proteomicist can understand which, if any, of  
217 their replicates are biasing the downstream results. As such, we consider PeptideMind to be a  
218 useful first step in the process of data validation prior to subsequent experiments. Consider the  
219 XP\_010027979.1 metabolic protein from Figure 3b. If an additional analysis or experiment  
220 determined this protein was biologically significant within our system, we would also be able to  
221 point to this Isolation Forest result as an indicator of statistical significance that falls within  
222 expected parameters. Conversely, we would consider the K1C9 protein in Figure 3a as showing  
223 too much variance for realistic application of statistical measures of significance. Biological  
224 conclusions that rely on such a protein would be considered putative until further orthogonal  
225 validation experiments were performed.

226  
227 At present, PeptideMind is not intended as a replacement for orthogonal protein validation  
228 experiments such as Western blotting, Parallel reaction monitoring, or other orthogonal mass  
229 spectrometry experiments. Rather, PeptideMind should serve as another tool in the proteomics  
230 toolbox which can be used to provide extra rigour for results, and demonstrate the validity of  
231 underlying quantitation without the need for additional experiments. The biggest hurdle with the

232 design of this program, however, is in the requirement for two states to have six replicates each.  
233 As replicate costs can be burdensome, PeptideMind is recommended for experiments in the  
234 discovery phase where tissue can be sourced relatively cheaply. In so doing, the researcher can  
235 accumulate a solid set of data backed up by PeptideMind, and the judicious use of other statistical  
236 measures, to narrow down their list of protein identifiers for further analysis.

237 PeptideMind is still in development phase, and there is much room for improvement. Some  
238 future additions may include:

- 239 1. Extra permutation potential when 7, 8, 9 or more replicates are specified per state
- 240 2. Additional MLAs incorporated into the replicate bias analysis
- 241 3. The incorporation of a python virtual environment for cleaner code production
- 242 4. Better real-time feedback to the user to update what stage the program has reached.

243 We hope that PeptideMind may serve as the inspiration for future experimentation and software  
244 development that leverages the power of permutations with MLAs for better, more specifically  
245 tailored analysis of differentially expressed proteins in proteomics experiments.

246

## 247 **5. Conclusions**

248 In this report, we have demonstrated the capabilities of the PeptideMind software in providing  
249 a valuable tool of statistical validation for data analysis pipelines in shotgun proteomics  
250 experiments. Leveraging the power of MLAs with permutation analysis, PeptideMind is capable  
251 of generating simple yet powerful graphical metrics whereby the user can assess the quality of  
252 their replicates, differential expression profiles, and resulting quantitation. In the future, we hope  
253 to expand the capability of the platform to incorporate further improvements and additional  
254 features.

255 **Conflict of Interest**

256 We confirm that there are no known conflicts of interest associated with this publication and  
257 there has been no significant financial support for this work that could have influenced its  
258 outcome.

259

260 **Acknowledgements**

261 PAH acknowledges Mike De Iuliis for continued support and encouragement. This work was  
262 supported by Macquarie University and the Biomolecular Discovery Research Centre.

263

264 **References**

- 265 1. Handler DC, Pascovici D, Mirzaei M, Gupta V, Salekdeh GH, Haynes PA. The Art of  
266 Validating Quantitative Proteomics Data. *Proteomics*. 2018;18(23):e1800222.
- 267 2. Kall L, Canterbury JD, Weston J, Noble WS, MacCoss MJ. Semi-supervised learning for  
268 peptide identification from shotgun proteomics datasets. *Nat Methods*. 2007;4(11):923-  
269 925.
- 270 3. Barla A, Jurman G, Riccadonna S, Merler S, Chierici M, Furlanello C. Machine learning  
271 methods for predictive proteomics. *Brief Bioinform*. 2008;9(2):119-128.
- 272 4. Elias JE, Gibbons FD, King OD, Roth FP, Gygi SP. Intensity-based protein identification by  
273 machine learning from a library of tandem mass spectra. *Nat Biotechnol*.  
274 2004;22(2):214-219.
- 275 5. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M,  
276 Prettenhofer P, Weiss R, Dubourg V. Scikit-learn: Machine learning in Python. *Journal of*  
277 *machine learning research*. 2011;12(Oct):2825-2830.
- 278 6. Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for  
279 balancing machine learning training data. *ACM SIGKDD explorations newsletter*.  
280 2004;6(1):20-29.
- 281 7. Handler DC, Haynes PA. An experimentally-derived measure of inter-replicate variation  
282 in reference samples: the same-same permutation methodology. *bioRxiv*. 2019:p.  
283 797217.
- 284 8. Neilson KA, Ali NA, Muralidharan S, Mirzaei M, Mariani M, Assadourian G, Lee A, van  
285 Sluyter SC, Haynes PA. Less label, more free: approaches in label-free quantitative mass  
286 spectrometry. *Proteomics*. 2011;11(4):535-553.
- 287 9. Zybailov B, Mosley AL, Sardiu ME, Coleman MK, Florens L, Washburn MP. Statistical  
288 analysis of membrane proteome expression changes in *Saccharomyces cerevisiae*. *J*  
289 *Proteome Res*. 2006;5(9):2339-2347.

- 290 10. Colaert N, Barsnes H, Vaudel M, Helsens K, Timmerman E, Sickmann A, Gevaert K,  
291 Martens L. Thermo-msf-parser: an open source Java library to parse and visualize  
292 Thermo Proteome Discoverer msf files. *J Proteome Res.* 2011;10(8):3840-3843.
- 293 11. Solntsev SK, Shortreed MR, Frey BL, Smith LM. Enhanced Global Post-translational  
294 Modification Discovery with MetaMorpheus. *J Proteome Res.* 2018;17(5):1844-1851.
- 295 12. Craig R, Beavis RC. TANDEM: matching proteins with tandem mass spectra.  
296 *Bioinformatics.* 2004;20(9):1466-1467.
- 297 13. Handler DCL, Cheng F, Shathili AM, Haynes PA. PeptideWitch – a software package to  
298 produce high-stringency proteomics data visualizations from label-free shotgun  
299 proteomics data. *Proteomes.* 2020;In Press.
- 300 14. Neilson KA, Keighley T, Pascovici D, Cooke B, Haynes PA. Label-free quantitative shotgun  
301 proteomics using normalized spectral abundance factors. *Methods Mol Biol.*  
302 2013;1002:205-222.

303

304

### 305 **Figure Legends**

306

307 *Figure 1:* Graphical User Interface which allows interaction with the PeptideMind software.

308 *Figure 2:* schematic diagram illustrating the replica permutation processing employed by  
309 PeptideMind.

310 *Figure 3:* Isolation Forests generated by PeptideMind for two selected proteins from an  
311 experiment comparing the proteome of young and old eucalyptus leaves. (A) K1C9, a human  
312 keratin protein present at variable levels, (B) XP\_010027978.1, a serine  
313 hydroxymethyltransferase metabolic protein.

314 *Figure 4:* Histograms of the average output values found from the MLAs used by PeptideMind.  
315 (A) data from an experiment comparing the proteome of young (replicates R1-R6) and old  
316 (replicates R7-R12) eucalyptus leaves, (B) data from an experiment comparing the proteome of  
317 two laboratory yeast strains known as CCB (replicates R1-R6) and CCC (replicates R7-R12).

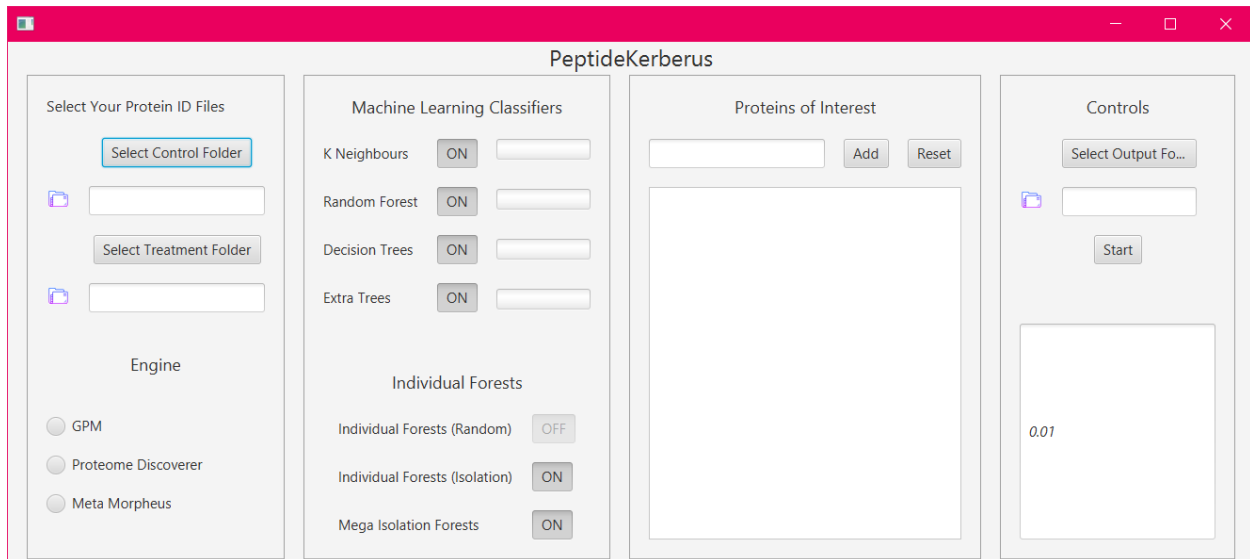
318

319

320

321

322 Figure 1



323

324

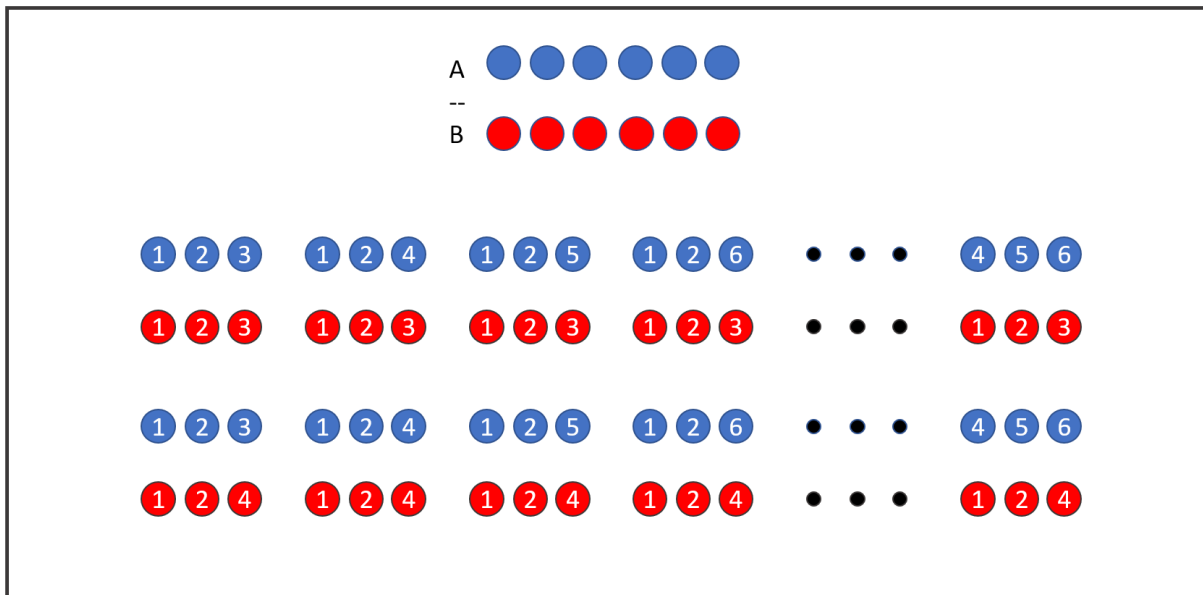
325 *Figure 1: Graphical User Interface which allows interaction with the PeptideMind software.*

326

327

328 Figure 2

Six replicates give us 400 non-redundant triplet comparisons



329

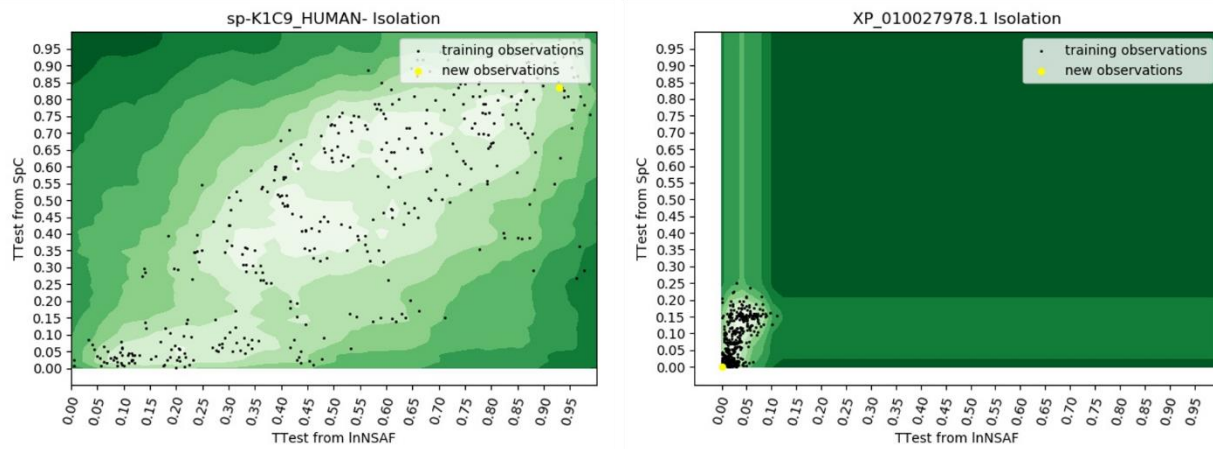
330

331 *Figure 2:* schematic diagram illustrating the replica permutation processing employed by  
332 PeptideMind.

333

334 Figure 3a (left) and 3b (right)

## Isolation Forest



335

336 *Figure 3:* Isolation Forests generated by PeptideMind for two selected proteins from an  
337 experiment comparing the proteome of young and old eucalyptus leaves. (A) K1C9, a human  
338 keratin protein present at variable levels, (B) XP\_010027978.1, a serine  
339 hydroxymethyltransferase metabolic protein.

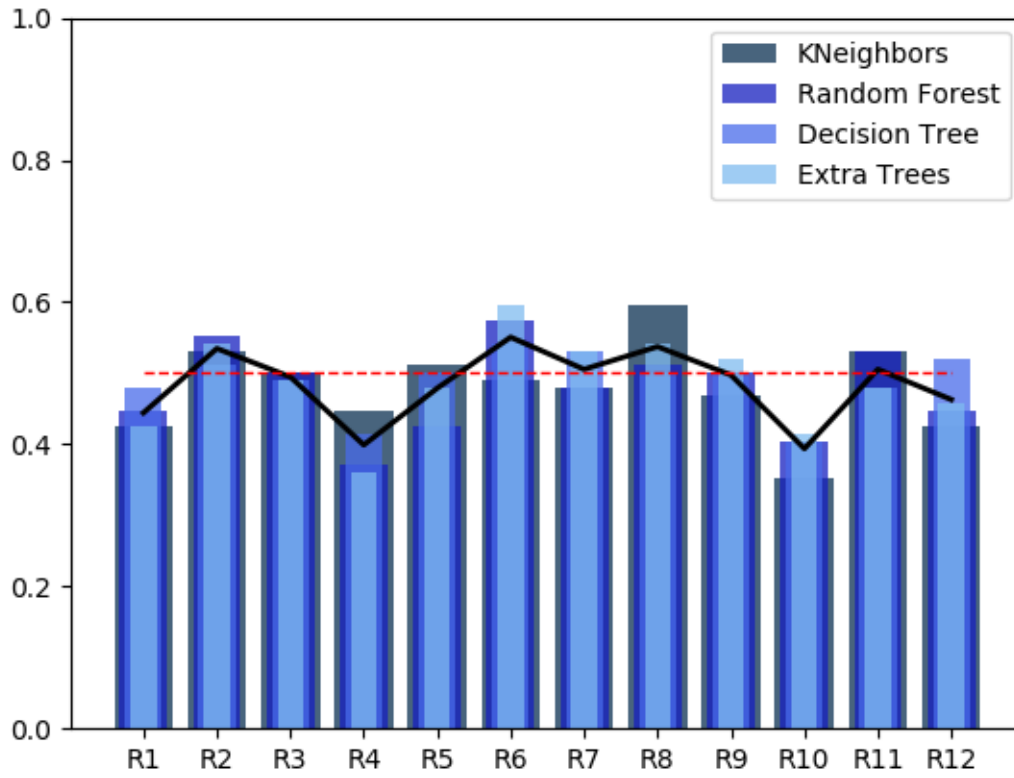
340

341

342



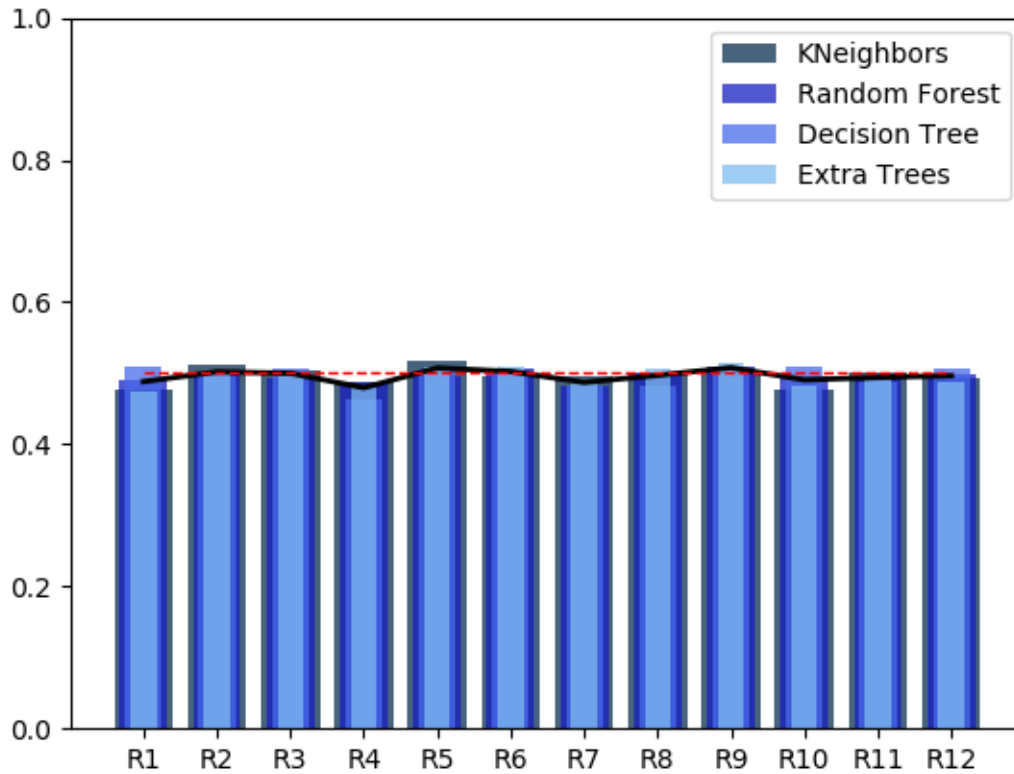
343 Figure 4a



344

345

346 Figure 4b



347

348

349 *Figure 4: Histograms of the average output values found from the MLAs used by PeptideMind.*  
350 (A) data from an experiment comparing the proteome of young (replicates R1-R6) and old  
351 (replicates R7-R12) eucalyptus leaves, (B) data from an experiment comparing the proteome of  
352 two laboratory yeast strains known as CCB (replicates R1-R6) and CCC (replicates R7-R12).

353