

isobar for quantification of PTM datasets

Florian P. Breitwieser, Jacques Colinge

March 23, 2016

Contents

1 Introduction

isobar [?] version 2 includes modules to facilitate PTM quantification. This vignette describes its parts, and how to use it to generate quantification reports.

```
> library(isobar) ## load the isobar package
```

Using *isobar*, automatic report generation is straight-forward given proper input files using the script `report/create_reports.R`. When called, it parses the global properties file `report/properties.R` and then the `properties.R` in the current directory. Below is a small example `properties.R` for creating a PDF Quality Control and XLSX analysis report:

```
type="iTRAQ4plexSpectra"
## peaklist files for quantitation, by default all mgf file in directory
peaklist=list.files(pattern="*\\.mgf$")

## id files, by default all id.csv files in directory
identifications=list.files(pattern="*\\.id.csv$")

modif="PHOS" # modification to track (eg PHOS, ACET, MET)
ptm.info.f <- getPtmInfoFromNextprot
spreadsheet.format="xlsx"
```

Reports will be generated calling `path_to_isobar/report/create_reports.R -peptide` from the directory containing the peaklists, identifications and `properties.R`.

2 Modification Site Localization

isobar supports PhosphoRS [?] and Delta Score [?] for modification site localization.

PhosphoRS integration The standalone Java version of PhosphoRS can be downloaded from <http://cores.imp.ac.at/uploads/media/PhosphoRS.zip>. It features a command line interface to a script which rescores localizations of the modification for each peptide-spectrum match. It uses XML files for input and output, which can be generated and parsed by **isobar**.

```
> # Generate PhosphoRS XML input file based on MGF and identification file
> # massTolerance: fragment ion mass tolerance (in Da)
> # activationType: CID, HCD, or ETD
> writePhosphoRSInput("phosphors.in.xml",
+                     "identifications.id.csv", "peaklist.mgf",
+                     massTolerance=0.5, activationType="CID")
```

After calling PhosphoRS (`java -jar phosphoRS.jar phosphors.in.xml phosphors.out.xml`), the resulting XML file can be read:

```
> # Read PhosphoRS XML output file
> # simplify: if TRUE, a data.frame is returned, else a list
> # besthit.only: if TRUE, only the best localization per spectrum is returned
> readPhosphoRSOutput("phosphors.out.xml", simplify=TRUE, besthit.only=TRUE)
```

`getPhosphoRSProbabilities` is a convenience function calling the writer, the script, and the reader in succession.

```
> getPhosphoRSProbabilities("identifications.id.csv", "peaklist.mgf",
+                           massTolerance=0.5, activationType="CID",
+                           phosphors.cmd="java -jar phosphoRS.jar")
```

Delta Score calculation The Mascot Delta Score can be calculated directly by the parser `mascotParser2.pl` and thresholded (*e. g.* `-minDeltaScore=10`). For CSV identification files which contain all hits for each spectrum (not just the best one), the function `calc.delta.score` within the R package is provided.

Using PhosphoRS and Delta Score in Report Generation. When generating an `IB-Spectra` object from `peaklist` and `identifications`, via `readIBSpectra`'s argument `annotate.spectra.f` a function can be plugged in to extend or modify the identification information. This can be used to calculate scores and filter localization scores with `filterSpectraDeltaScore`) or `annotateSpectraPhosphoRS`.

```
> # filterSpectraDeltaScore calls calc.delta.score
> # if no column named delta.score is present in the data frame
> # identifications below a min.delta.score are REMOVED
> ib <- readIBSpectra("identifications.id.csv", "peaklist.mgf",
+                    annotate.spectra.f=function(...)
+                    filterSpectraDeltaScore(..., min.delta.score=10))
> # filterSpectraPhosphoRS calls PhosphoRS to calculate PhosphoRS probabilities
> # identifications below a min.prob (PhosphoRS peptide isoform probability)
```

[illegible]

This can be used in report generation, too, where the `readIBSpectra.args` can be set accordingly in the report properties file `properties.R`:

```
readIBSpectra.args = list(annotate.spectra.f=filterSpectraDeltaScore)
```

or

```
readIBSpectra.args = list(annotate.spectra.f=filterSpectraPhosphoRS)
```

3 Peptide Ratio Calculation

All functions which are available to calculate ratios on protein level can also be used for peptides. The same noise model is appropriate for both.

```
> data(ib_phospho)
> data(noise.model.hcd)
> head(proteinGroup(ib_phospho)@peptideInfo)
```

	protein	peptide	start_pos		modif
2072	A1L390-1	SPLSPTETFSWPDVR	1037		
2074	A1L390-2	SPLSPTETFSWPDVR	570		
2076	A1L390-3	SPLSPTETFSWPDVR	981		
1299	A6NKT7	LLLDLPLQTPHK	1170		
783	O00264	GDQPAASGDSDDDEPPPLPR	48		
2045	O14497-1	SPFLHSGMK	1604		
2072	iTRAQ4plex_Nterm:PHOS::PHOS::::::::::::				
2074	iTRAQ4plex_Nterm:PHOS::PHOS::::::::::::				
2076	iTRAQ4plex_Nterm:PHOS::PHOS::::::::::::				
1299	iTRAQ4plex_Nterm:::::::::PHOS::iTRAQ4plex_K:				
783	iTRAQ4plex_Nterm:::::::::PHOS::::::::::::				
2045	iTRAQ4plex_Nterm:::::::::PHOS::iTRAQ4plex_K:				

```
> 10^estimateRatio(ib_phospho,noise.model.hcd,peptide="SPLSPTETFSWPDVR")
```

	114	115	116	117
114	1.0000000	0.3088721	1.4354943	1.641885
115	3.2375859	1.0000000	4.6497966	5.318776
116	0.6966241	0.2150632	1.0000000	1.143867
117	0.6090561	0.1880132	0.8742276	1.000000

By giving a matrix to `estimateRatio`, we can calculate ratios for peptides with specific modifications:

```
> pep.n.modif <- unique(apply(fData(ib_phospho)[,c("peptide","modif")],2,cbind))
> print(head(pep.n.modif))

      peptide
[1,] "AAATPESQEPQAK"
[2,] "AAEAGGAEEQYGFLTTPTK"
[3,] "AAEEQGDDQDSEK"
[4,] "AAPPPGSPAK"
[5,] "AAVGQESPGGLEAGNAK"
[6,] "AAVLSDSEDEEK"
      modif
[1,] "iTRAQ4plex_Nterm:::::PHOS:::::iTRAQ4plex_K:"
[2,] "iTRAQ4plex_Nterm:::::PHOS:::::iTRAQ4plex_K:"
[3,] "iTRAQ4plex_Nterm:::::PHOS::iTRAQ4plex_K:"
[4,] "iTRAQ4plex_Nterm:::::PHOS::iTRAQ4plex_K:"
[5,] "iTRAQ4plex_Nterm:::::PHOS:::::iTRAQ4plex_K:"
[6,] "iTRAQ4plex_Nterm:::::PHOS::PHOS:::::iTRAQ4plex_K:"

> estimateRatio(ib_phospho,noise.model.hcd,channel1="114",channel2="115",
+               peptide=head(pep.n.modif),combine=FALSE)[,c("lratio","variance",
+               "n.spectra","p.value.rat")]

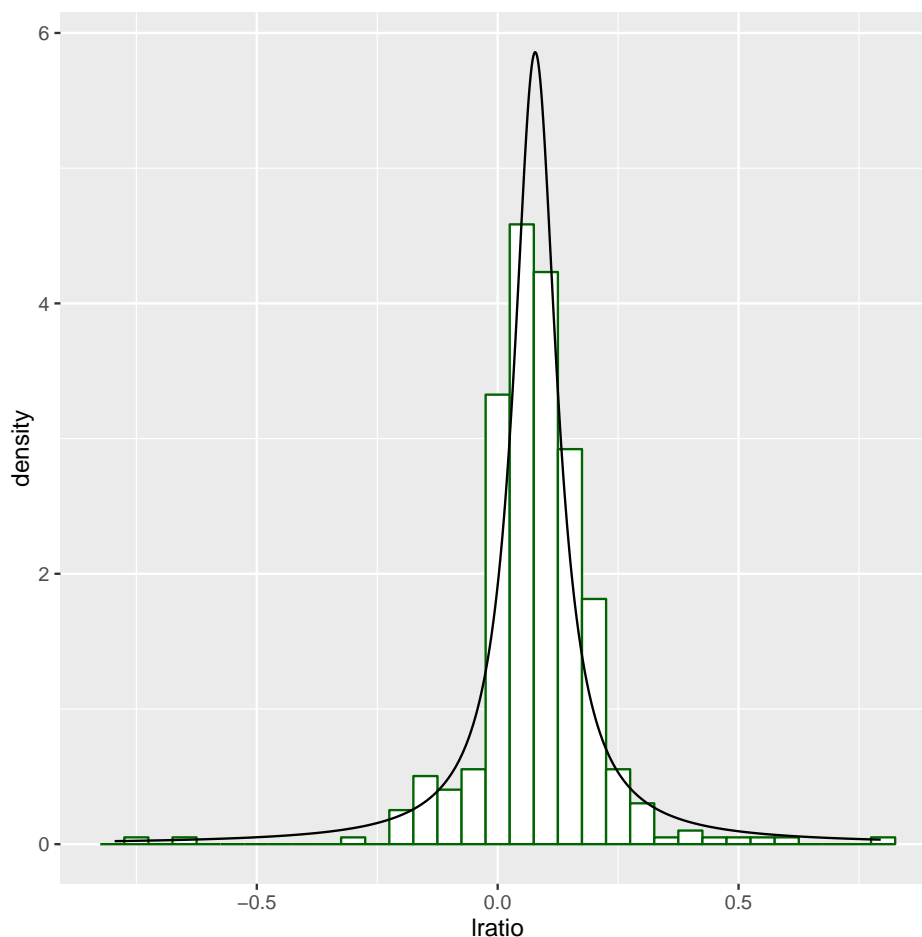
      lratio  variance n.spectra  p.value.rat
1 -0.6978020 0.01034090         2 3.394310e-12
2          NaN         Inf         0          NA
3  0.1388425 0.01052788         2 8.800084e-02
4 -1.0793665 0.04166971         1 6.196545e-08
5 -0.9655771 0.02406589         1 2.419553e-10
6 -0.2164083 0.08305527         7 2.263522e-01

>
```

A ratio distribution can be calculated based on peptide ratios:

```
> suppressPackageStartupMessages(library(distr))
> suppressPackageStartupMessages(library(ggplot2))
> peptide.ratios <- peptideRatios(ib_phospho,noise.model=noise.model.hcd,
+                               cmbn=matrix(c("114","116"),ncol=1))
> lim <- max(abs(peptide.ratios$lratio),na.rm=TRUE)
> peptide.distr.cauchy <- fitCauchy(peptide.ratios$lratio)
> pseq <- seq(from=-lim,to=lim,length.out=1000)
> ggplot() +
+   geom_histogram(aes(x=lratio,y=..density..),data=peptide.ratios,binwidth=0.05,
```

```
+           color="darkgreen",fill="white") +
+   geom_line(aes(x=x,y=y),color="black",
+             data=data.frame(x=pseq,y=d(peptide.distr.cauchy)(pseq)))
```



Correction with protein ratios. The observed change in concentration of modified peptides in one condition versus another is integrating two separate modes of regulation [?]:

1. Protein expression change
2. Modification state change

In many cases, it thus can be advisable to conduct separate MS quantification runs of the peptides enriched for the modification of interest, AND the global proteome quantification.

In the report generation, data from other experiments can be integrated using the property `compare.to.quant` in `properties.R`:

```
load("../proteome/quant.tbl.rda")           # load proteome quantification table
compare.to.quant=list(proteome=quant.tbl) # set property
rm(quant.tbl)
```

Peptide ratios can also be corrected with proteome ratios of a separate experiment, when giving as `peptide` argument a `matrix` or `data.frame` with columns for 'peptide', 'modif', and 'correct.ratio'. 'correct.ratio' is a \log_{10} ratio which will be used to adjust the one calculated for the specific modified peptide.

```
> peptides <- pep.n.modif[1:5,]
> orig.ratio <- estimateRatio(ib_phospho,noise.model.hcd,channel1="114",channel2="115",
+                             peptide=peptides,combine=FALSE)[,c("lratio","variance")]
> peptides.c <- cbind(peptides,correct.ratio=c(0,-1,1,2,-2))
> corr.ratio <- estimateRatio(ib_phospho,noise.model.hcd,channel1="114",channel2="115",
+                             peptide=peptides.c,combine=FALSE)[,c("lratio","variance")]
> data.frame(peptides.c,orig.ratio,corr.ratio)
```

	peptide		peptide		modif
1	AAATPESQEPQAK		iTRAQ4plex_Nterm:::PHOS:::iTRAQ4plex_K:		
2	AAEAGGAEEQYGFLLTPTK		iTRAQ4plex_Nterm:::PHOS:::iTRAQ4plex_K:		
3	AAEEQGDDQDSEK		iTRAQ4plex_Nterm:::PHOS:::iTRAQ4plex_K:		
4	AAPPPGSPAK		iTRAQ4plex_Nterm:::PHOS:::iTRAQ4plex_K:		
5	AAVGQESPGGLEAGNAK		iTRAQ4plex_Nterm:::PHOS:::iTRAQ4plex_K:		
	correct.ratio	lratio	variance	lratio.1	variance.1
1	0	-0.6978020	0.01034090	-0.3736099	0.1962761
2	-1	NaN	Inf	-0.3736099	0.1962761
3	1	0.1388425	0.01052788	-0.3736099	0.1962761
4	2	-1.0793665	0.04166971	-0.3736099	0.1962761
5	-2	-0.9655771	0.02406589	-0.3736099	0.1962761

As apparent, the variance stays the same also for corrected ratios. If a fourth column `variance` of the `peptide` argument reports the variance of the correction ratio, it is added to the calculated ratio's variance (assuming independence).

4 Harvesting public PTM databases

neXtProt [?] and PhosphoSitePlus [?] provide information on experimentally determined post-translational modifications. neXtProt focuses on man, and PhosphoSitePlus on man and mouse. Both are manually curated and annotate thousands of residues of post-translationally modified proteins.

`isobar` provides functions to gather their information on identified proteins.

```
> ptm.info <- getPtmInfoFromPhosphoSitePlus(proteinGroup(ib_phospho),modif="PHOS")
> ptm.info <- getPtmInfoFromNextprot(proteinGroup(ib_phospho))

> head(ptm.info)
```

	.id	isoform_ac	quality	description	evidence	first_position	last_position
1	A1L390	A1L390-1	GOLD	Phosphoserine	EXP	76	76
2	A1L390	A1L390-1	SILVER	Phosphoserine	EXP	433	433

3	A1L390	A1L390-1	GOLD	Phosphoserine	Curated	533	533
4	A1L390	A1L390-1	GOLD	Phosphoserine	EXP	576	576
5	A1L390	A1L390-1	GOLD	Phosphoserine	EXP	577	577
6	A1L390	A1L390-1	SILVER	Phosphoserine	EXP	614	614
	modification.name	modification.accession	position				
1	Phosphoserine	PTM-0253	76				
2	Phosphoserine	PTM-0253	433				
3	Phosphoserine	PTM-0253	533				
4	Phosphoserine	PTM-0253	576				
5	Phosphoserine	PTM-0253	577				
6	Phosphoserine	PTM-0253	614				

For reports, the function can be selected via the property `ptm.info.f` in `properties.R`:

```
protein.info.f = getPtmInfoFromNextprot
```

For PhosphoSitePlus, define the modification to get the correct dataset:

```
ptm.info.f <- function(...) getPtmInfoFromPhosphoSitePlus(...,modification="PHOS")
```

PhosphoSitePlus datasets will be downloaded from their website to 'Phosphorylation_site_dataset.gz' or 'Acetylation_site_dataset.gz', etc (see mapping property of `getPtmInfoFromPhosphoSitePlus`) unless a file with that name exists.