

# Zebrafish Anatomical Term Enrichment

# ZFIN annotation

- As well as annotating genes with GO terms, ZFIN associate genes with anatomical terms (based on literature review and *in situ* images submitted directly)
- For example, rhodopsin (<https://zfin.org/ZDB-GENE-990415-271>)

Term ▼	Annotation Extension	Evidence	With/From	Citations
<a href="#">absorption of visible light</a>		IDA		<a href="#">Tartelin et al., 2011</a>
<a href="#">detection of light stimulus</a>		IDA		<a href="#">Morrow et al., 2015</a>

Expression Location ▼	Stage Observed					Citations
	0 hpf	3.33 hpf	10.33 hpf	60 hpf	90 dpf	
<a href="#">brain</a>						<a href="#">Davies et al., 2015</a>
<a href="#">epiphysis</a>						(2)
<a href="#">epiphysis photoreceptor cell</a>						<a href="#">Schredelseker et al., 2018</a>

# Anatomical Hierarchy

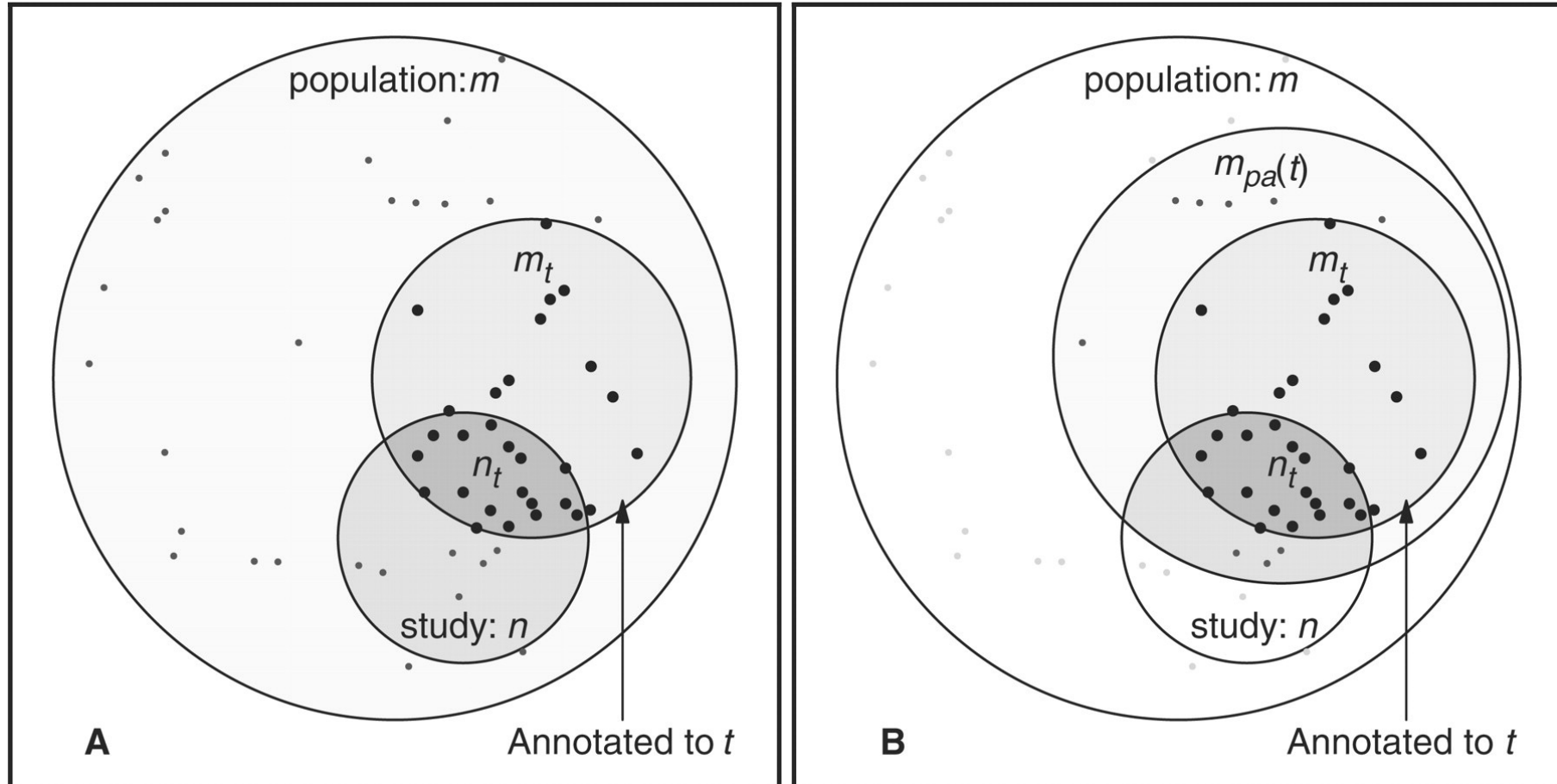
- The zebrafish anatomy (ZFA) terms form a tree-like hierarchy, where each term can have children and parents

```
* ZFA:0009127 photoreceptor cell
  * ZFA:0009154 eye photoreceptor cell
    * ZFA:0009262 retinal cone cell
    * ZFA:0009275 retinal rod cell
  * ZFA:0009219 visible light photoreceptor cell
    * ZFA:0009220 photopic photoreceptor cell
      * ZFA:0009222 blue sensitive photoreceptor cell
      * ZFA:0009223 green sensitive photoreceptor cell
      * ZFA:0009224 red sensitive photoreceptor cell
    * ZFA:0009221 UV sensitive photoreceptor cell
```

# Anatomical Enrichment

- To check for enrichment of terms, can't just check each term as if they are independent of each other
  - They aren't, so you'll get spurious enrichments for some child terms of terms that are significantly enriched
- Instead check for enrichment of a term in the context of the genes annotated to the term's parents

# Anatomical Enrichment



From Grossman et al, 2007

# Files

- Need `zfin.txt` file linking ZFIN gene IDs to ZFA terms:  
<https://funcgen2022.buschlab.org/downloads/zfin.txt>
- And `zfa.obo` file describing all ZFA terms:  
<https://funcgen2022.buschlab.org/downloads/zfa.obo>
- <http://ontologizer.de/> is software for testing enrichment of ontology terms
- Need `Ontologizer.jar` file:  
<https://funcgen2022.buschlab.org/downloads/Ontologizer.jar>
- All files are also available on "penelopeprime"

# How Files Made

- `wget https://raw.githubusercontent.com/cerivs/zebrafish-anatomical-ontology/master/zfa.obo`
- `wget http://zfin.org/downloads/phenoGeneCleanData_fish.txt`
- `wget http://zfin.org/downloads/wildtype-expression_fish.txt`
- `wget http://ontologizer.de/cmdline/Ontologizer.jar`
- `cut -f3,4,8,13,17 phenoGeneCleanData_fish.txt | grep ZDB-GENE- | awk '{ print $1 "\t" $2 "\n" $1 "\t"-$3 "\n" $1 "\t" $4 "\n" $1 "\t" $5 }' | grep ZFA: > zfin1.tmp`
- `cut -f1,4,6 wildtype-expression_fish.txt | grep ZDB-GENE- | awk '{ print $1 "\t" $2 "\n" $1 "\t" $3 }' | grep ZFA: > zfin2.tmp`
- `sort -u zfin1.tmp zfin2.tmp | awk '{ print "ZFIN\t" $1 "\t" $1 "\t\t" $2 "\tRef\tND\tC\t" $1 "\t\tgene_product\ttaxon:7955\t20220929\tZFIN" }' > zfin.txt`
- `rm phenoGeneCleanData_fish.txt wildtype-expression_fish.txt zfin1.tmp zfin2.tmp`

# More Files

- Also need a file listing all the detectable Ensembl IDs (the population set):

```
awk -F"\t" '$3 != "NA"' Amp.counts.tsv | cut -f1 | grep ENS > population.tsv
```

- And a file listing all the significant Ensembl IDs (the study set):

```
awk -F"\t" '$3 < 0.05' Amp.counts.tsv | cut -f1 | grep ENS > study.tsv
```

- Then use BioMart to convert Ensembl IDs to ZFIN IDs to make `population.zfin.tsv` and `study.zfin.tsv`



# Best population set?

- Population set should be all the genes that are detectable
- Do NOT just use all the genes in the genome (which some online tools default to)
- Instead either:
  - Use all the genes that are above some threshold number of counts (with 1 count not being unreasonable)
  - If you're using DESeq2, use all the genes that haven't been subjected to independent filtering (i.e. those that don't have "NA" for their adjusted p-value)
  - See <http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#independent-filtering-of-results> for more info on independent filtering

# Run Ontologizer

- `java -jar Ontologizer.jar -a zfin.txt -g zfa.obo -c Parent-Child-Union -m Bonferroni -p population.zfin.tsv -s study.zfin.tsv`
- **Will produce a file called `table-study.zfin-Parent-Child-Union-Bonferroni.txt` containing all the ZFA terms, including significantly enriched ones**

# Ontologizer Output

- `cut -f1,11,13 table-study.zfin-Parent-Child-Union-Bonferroni.txt | awk '$2 < 0.05'`

```
ZFA:0000368      0.012688503803582591      "integument"  
ZFA:0000036      0.01685320424580877      "liver and biliary system"  
ZFA:0000339      0.04210104402183481      "digestive system"  
ZFA:0000123      0.04976546826621127      "liver"  
ZFA:0001491      0.04976546826621127      "solid compound organ"
```

# Thank You

Any Questions?