Meta'omic functional profiling with HUMAnN

Curtis Huttenhower

01-19-15

Harvard School of Public Health
Department of Biostatistics

U. Oregon

The two big questions…

Who is there?

(taxonomic profiling)

What are they doing?

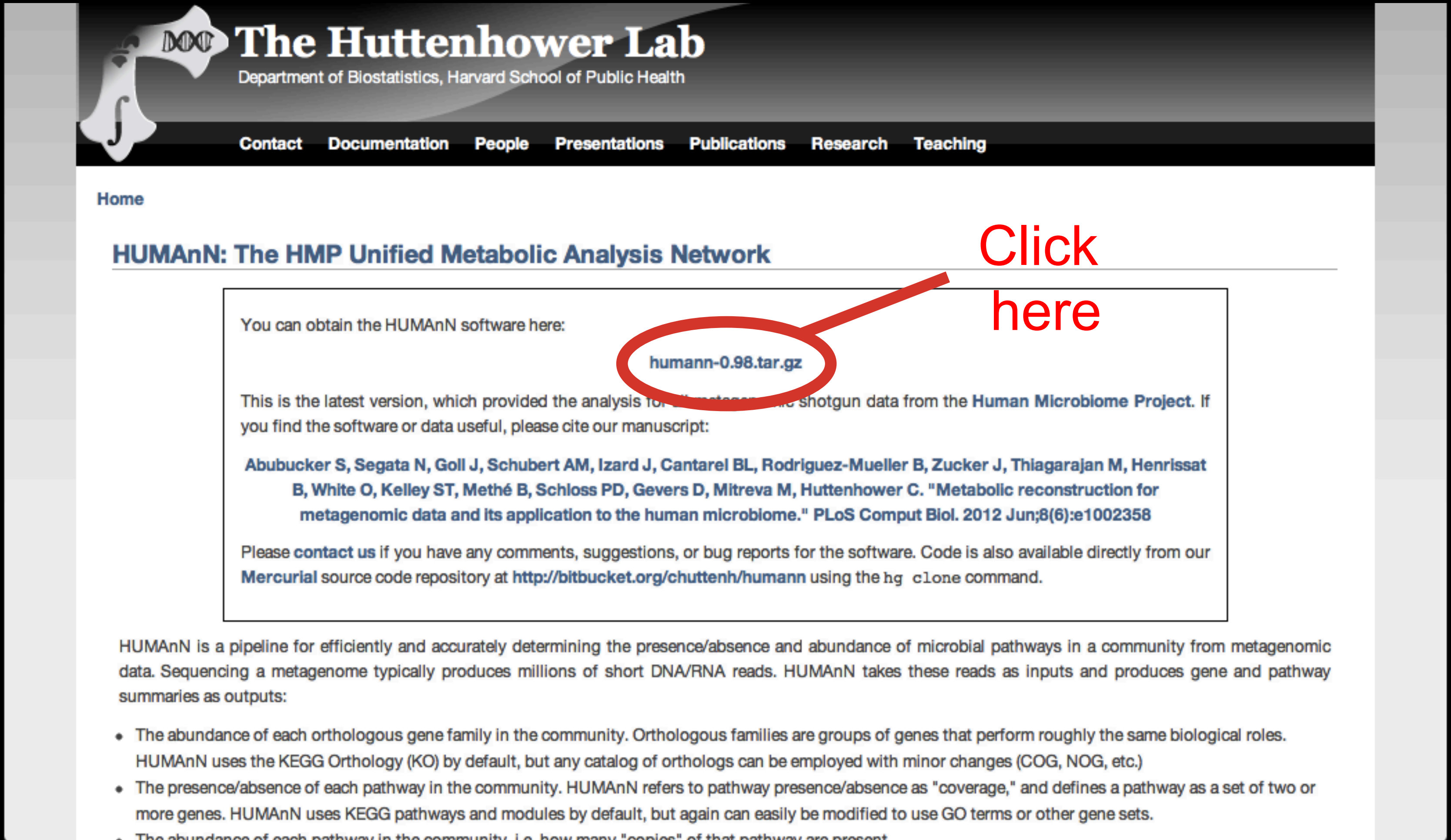
(functional profiling)

Setup notes reminder

- Slides with green titles or text include instructions not needed today, but useful for your own analyses
- Keep an eye out for red warnings of particular importance
- Command lines and program/file names appear in a **`monospaced font`**.
- Commands you should specifically copy/paste are in **`monospaced bold blue`**.

What they're doing: HUMAnN

- As a broad functional profiler, you *could* download HUMAnN at: http://huttenhower.sph.harvard.edu/humann



HUMAnN: The HMP Unified Metabolic Analysis Network

Click here

You can obtain the HUMAnN software here:

humann-0.98.tar.gz

This is the latest version, which provided the analysis fo [illegible] shotgun data from the **Human Microbiome Project**. If you find the software or data useful, please cite our manuscript:

Abubucker S, Segata N, Goll J, Schubert AM, Izard J, Cantarel BL, Rodriguez-Mueller B, Zucker J, Thiagarajan M, Henrissat B, White O, Kelley ST, Methé B, Schloss PD, Gevers D, Mitreva M, Huttenhower C. "Metabolic reconstruction for metagenomic data and its application to the human microbiome." PLoS Comput Biol. 2012 Jun;8(6):e1002358

Please **contact us** if you have any comments, suggestions, or bug reports for the software. Code is also available directly from our **Mercurial** source code repository at **http://bitbucket.org/chuttenh/humann** using the `hg clone` command.

HUMAnN is a pipeline for efficiently and accurately determining the presence/absence and abundance of microbial pathways in a community from metagenomic data. Sequencing a metagenome typically produces millions of short DNA/RNA reads. HUMAnN takes these reads as inputs and produces gene and pathway summaries as outputs:

- The abundance of each orthologous gene family in the community. Orthologous families are groups of genes that perform roughly the same biological roles. HUMAnN uses the KEGG Orthology (KO) by default, but any catalog of orthologs can be employed with minor changes (COG, NOG, etc.)
- The presence/absence of each pathway in the community. HUMAnN refers to pathway presence/absence as "coverage," and defines a pathway as a set of two or more genes. HUMAnN uses KEGG pathways and modules by default, but again can easily be modified to use GO terms or other gene sets.
- The abundance of each pathway in the community, i.e. how many "copies" of that pathway are present.

What they’re doing: HUMAnN

- ...but instead we’ve already downloaded it
- Expand HUMAnN (no install!)

```
Cd ~/workshop_data/metagenomics
tar -xzf ~/workshop_data/metagenomics/biobakery/software/humann-v0.99.tar.gz
```

- Set up a link to the KEGG reference DB:

```
ln -s ~/workshop_data/metagenomics/biobakery/data/kegg.reduced.udb
```

- And although you would normally download USEARCH from here:
  - http://www.drive5.com/usearch/download.html

  We’re going to use it preinstalled instead

What they're doing: HUMAnN

- If we weren't all running this, you'd need to:
  - Get KEGG – used to be free, now it's not!
    - Fortunately, we have a HUMAnN-compatible distributable version; contact me...
  - Index it for USEARCH:

```
usearch -makeudb_usearch kegg.reduced.fasta
   -output kegg.reduced.udb
```

- This takes a minute or two, so we've precomputed it; thus, forge ahead...

What they're doing: HUMAnN

- Did you notice that we didn't QC our data at all?
  - MetaPhlAn is very robust to junk sequence
  - HUMAnN is pretty robust, but not quite as much

- We've already run a standard metagenomic QC:
  - Quality trim by removing bad bases (typically Q ~15)
  - Length filter to remove short sequences (typically <75%)

What they're doing: HUMAnN

- Must start from FASTQ files to do this
- Quality trim by removing bad bases:

FASTX: `fastx_trimmer`
ea-utils: `fastq-mcf`

- Length filter by removing short sequences:
  - 75% of original length is standard (thus 75nt from 100nt reads)

FASTX: `fastx_quality_filter`
USEARCH: `fastq_filter`

- Now convert your FASTQ to a FASTA:

FASTX: `fastq_to_fasta`
USEARCH: `fastq_filter`

- Some final caveats:
  - If you're using paired end reads, match filters!
  - See my course homeworks at http://huttenhower.sph.harvard.edu/bio508
  - Aren't you glad you're not doing this today?

What they’re doing: HUMAnN

- Enter the humann directory and fetch the data

```
ln -s  ~/workshop_data/metagenomics/biobakery/data/763577454-SRS014459-Stool.fasta
cd humann-0.99
```

- Run your first translated BLAST search:

```
usearch
  -usearch_local ../763577454-SRS014459-Stool.fasta
  -db ../kegg.reduced.udb -id 0.8
  -blast6out input/763577454-SRS014459-Stool.txt
```

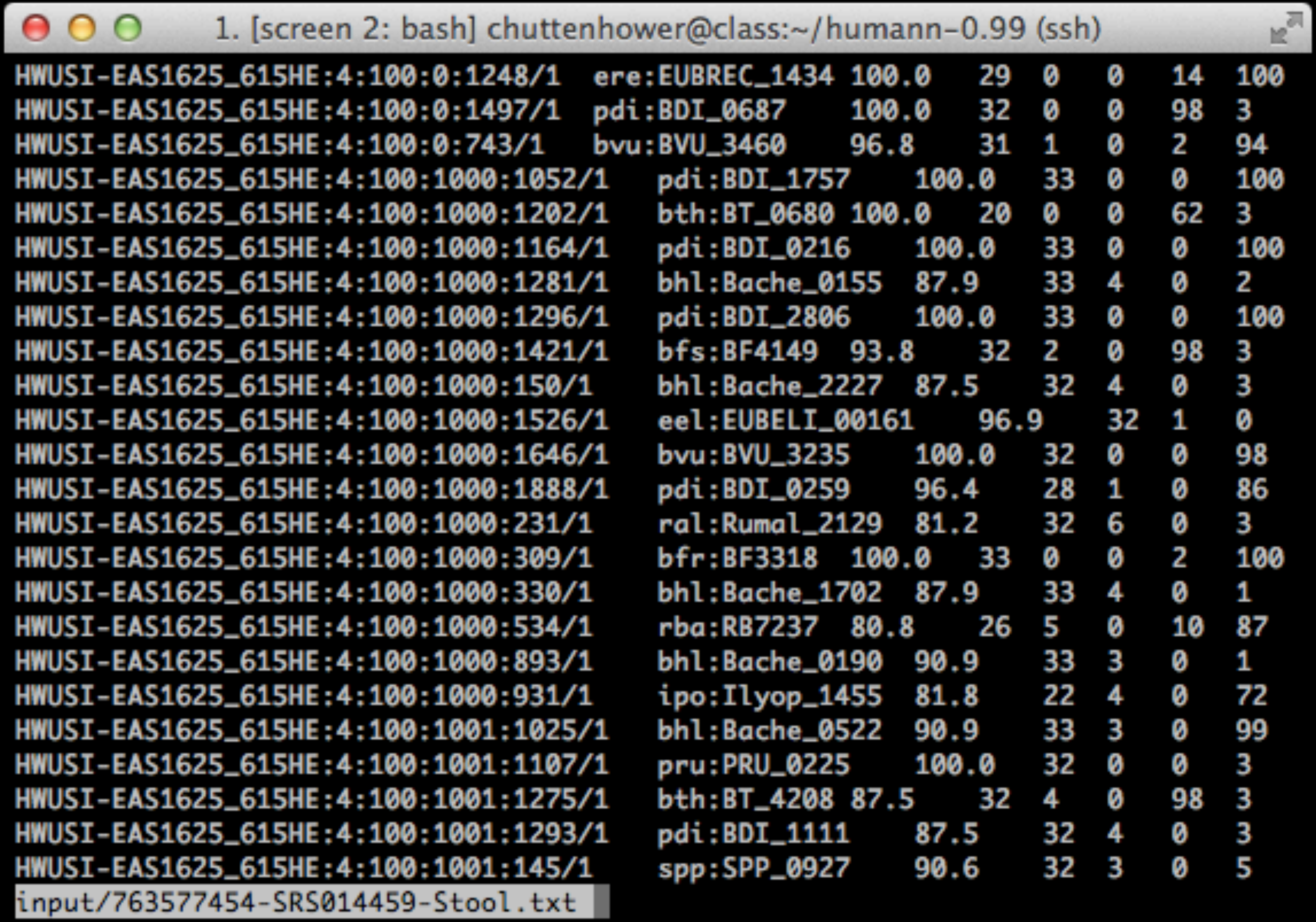
- What did you just do?

```
less input/763577454-SRS014459-Stool.txt
```

  - Recall BLAST’s tab-delimited output headers:
    - qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore

- **Rinse and repeat for the remaining samples**

What they're doing: HUMAnN



```
HWUSI-EAS1625_615HE:4:100:0:1248/1  ere:EUBREC_1434 100.0   29  0   0   14  100
HWUSI-EAS1625_615HE:4:100:0:1497/1  pdi:BDI_0687    100.0   32  0   0   98  3
HWUSI-EAS1625_615HE:4:100:0:743/1   bvu:BVU_3460    96.8    31  1   0   2   94
HWUSI-EAS1625_615HE:4:100:1000:1052/1   pdi:BDI_1757    100.0   33  0   0   100
HWUSI-EAS1625_615HE:4:100:1000:1202/1   bth:BT_0680 100.0   20  0   0   62  3
HWUSI-EAS1625_615HE:4:100:1000:1164/1   pdi:BDI_0216    100.0   33  0   0   100
HWUSI-EAS1625_615HE:4:100:1000:1281/1   bhl:Bache_0155  87.9    33  4   0   2
HWUSI-EAS1625_615HE:4:100:1000:1296/1   pdi:BDI_2806    100.0   33  0   0   100
HWUSI-EAS1625_615HE:4:100:1000:1421/1   bfs:BF4149  93.8    32  2   0   98  3
HWUSI-EAS1625_615HE:4:100:1000:150/1    bhl:Bache_2227  87.5    32  4   0   3
HWUSI-EAS1625_615HE:4:100:1000:1526/1   eel:EUBELI_00161    96.9    32  1   0
HWUSI-EAS1625_615HE:4:100:1000:1646/1   bvu:BVU_3235    100.0   32  0   0   98
HWUSI-EAS1625_615HE:4:100:1000:1888/1   pdi:BDI_0259    96.4    28  1   0   86
HWUSI-EAS1625_615HE:4:100:1000:231/1    ral:Rumal_2129  81.2    32  6   0   3
HWUSI-EAS1625_615HE:4:100:1000:309/1    bfr:BF3318  100.0   33  0   0   2   100
HWUSI-EAS1625_615HE:4:100:1000:330/1    bhl:Bache_1702  87.9    33  4   0   1
HWUSI-EAS1625_615HE:4:100:1000:534/1    rba:RB7237  80.8    26  5   0   10  87
HWUSI-EAS1625_615HE:4:100:1000:893/1    bhl:Bache_0190  90.9    33  3   0   1
HWUSI-EAS1625_615HE:4:100:1000:931/1    ipo:Ilyop_1455  81.8    22  4   0   72
HWUSI-EAS1625_615HE:4:100:1001:1025/1   bhl:Bache_0522  90.9    33  3   0   99
HWUSI-EAS1625_615HE:4:100:1001:1107/1   pru:PRU_0225    100.0   32  0   0   3
HWUSI-EAS1625_615HE:4:100:1001:1275/1   bth:BT_4208 87.5    32  4   0   98  3
HWUSI-EAS1625_615HE:4:100:1001:1293/1   pdi:BDI_1111    87.5    32  4   0   3
HWUSI-EAS1625_615HE:4:100:1001:145/1    spp:SPP_0927    90.6    32  3   0   5
input/763577454-SRS014459-Stool.txt
```

What they're doing: HUMAnN

- Normally you'd need to install SCons from:
  - http://www.scons.org

- Instead, we'll use it preinstalled as well, so...

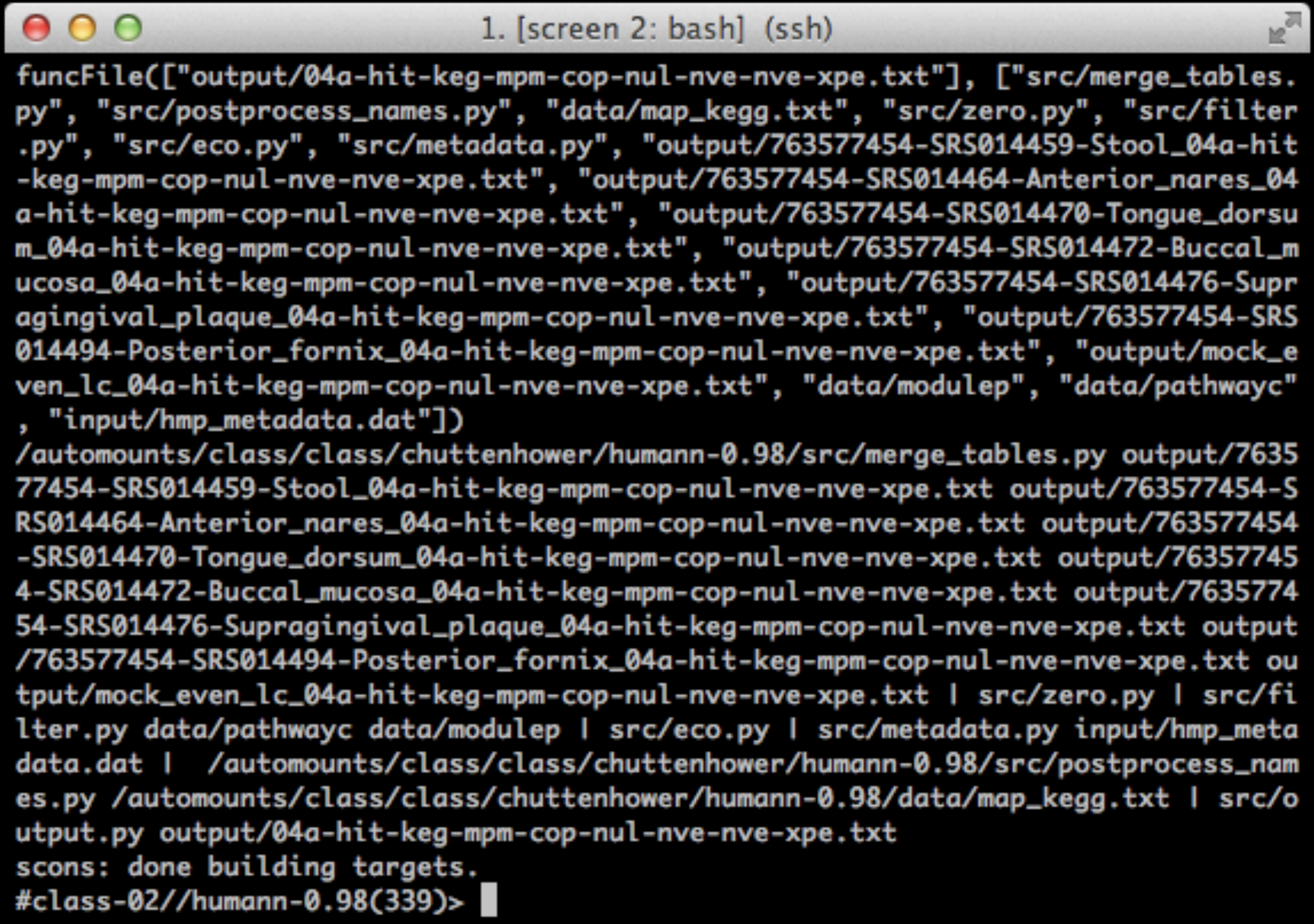
GO!

```
~/workshop_data/metagenomics/biobakery/software/ext/scons-2.3.2/bin/scons
```

- You should see a bunch of text scroll by
  - Note: you can run `scons -j8` to parallelize tasks

What they're doing: HUMAnN

- After a minute or two, you should see:



```
funcFile(["output/04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt"], ["src/merge_tables.
py", "src/postprocess_names.py", "data/map_kegg.txt", "src/zero.py", "src/filter
.py", "src/eco.py", "src/metadata.py", "output/763577454-SRS014459-Stool_04a-hit
-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/763577454-SRS014464-Anterior_nares_04
a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/763577454-SRS014470-Tongue_dorsu
m_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/763577454-SRS014472-Buccal_m
ucosa_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/763577454-SRS014476-Supr
agingival_plaque_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/763577454-SRS
014494-Posterior_fornix_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "output/mock_e
ven_lc_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt", "data/modulep", "data/pathwayc"
, "input/hmp_metadata.dat"])
/automounts/class/class/chuttenhower/humann-0.98/src/merge_tables.py output/7635
77454-SRS014459-Stool_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt output/763577454-S
RS014464-Anterior_nares_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt output/763577454
-SRS014470-Tongue_dorsum_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt output/76357745
4-SRS014472-Buccal_mucosa_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt output/7635774
54-SRS014476-Supragingival_plaque_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt output
/763577454-SRS014494-Posterior_fornix_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt ou
tput/mock_even_lc_04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt | src/zero.py | src/fi
lter.py data/pathwayc data/modulep | src/eco.py | src/metadata.py input/hmp_meta
data.dat |  /automounts/class/class/chuttenhower/humann-0.98/src/postprocess_nam
es.py /automounts/class/class/chuttenhower/humann-0.98/data/map_kegg.txt | src/o
utput.py output/04a-hit-keg-mpm-cop-nul-nve-nve-xpe.txt
scons: done building targets.
#class-02//humann-0.98(339)>
```

What they’re doing: HUMAnN

- This has created four main files:
  - Two each for pathways (big) and modules (small)
  - Two each for coverage and relative abundance
- Each is tab-delimited text with one column per sample
- All four are in the `output` directory:

output/04**a**-hit-keg-mp**t**-cop-nul-nve-nve-xpe.txt

  - Coverage (a) of pathways (t)

output/04**a**-hit-keg-mp**m**-cop-nul-nve-nve-xpe.txt

  - Coverage (a) of modules (m)

output/04**b**-hit-keg-mp**t**-cop-nul-nve-nve.txt

  - Abundance (b) of pathways (t)

output/04**b**-hit-keg-mp**m**-cop-nul-nve-nve.txt

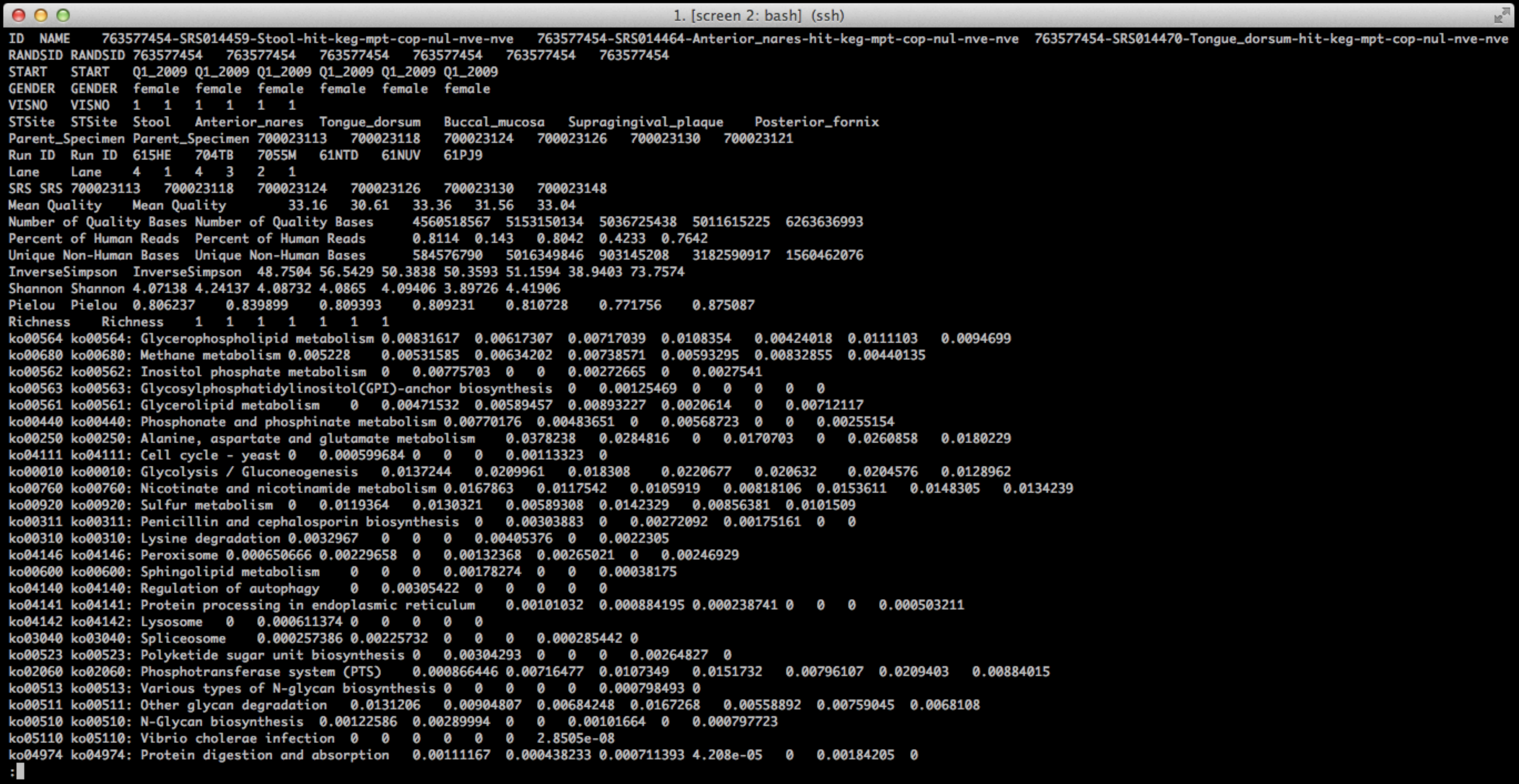
  - Abundance (b) of modules (m)

- I almost always just use `04b-mpm` (module abundances)

What they're doing: HUMAnN

- Let's take a look:

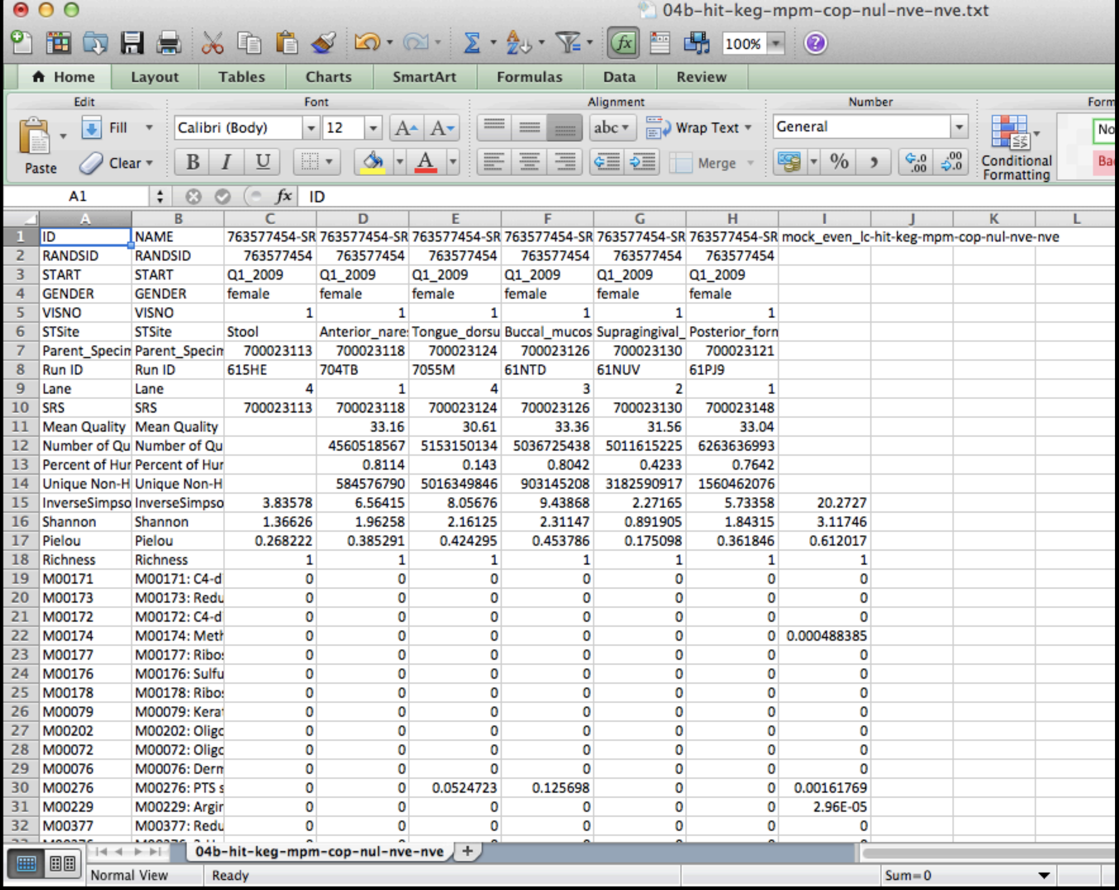
  `less output/04b-hit-keg-mpm-cop-nul-nve-nve.txt`



```
ID  NAME    763577454-SRS014459-Stool-hit-keg-mpt-cop-nul-nve-nve   763577454-SRS014464-Anterior_nares-hit-keg-mpt-cop-nul-nve-nve  763577454-SRS014470-Tongue_dorsum-hit-keg-mpt-cop-nul-nve-nve
RANDSID RANDSID 763577454   763577454   763577454   763577454   763577454   763577454
START   START   Q1_2009 Q1_2009 Q1_2009 Q1_2009 Q1_2009 Q1_2009
GENDER  GENDER  female  female  female  female  female  female
VISNO   VISNO   1   1   1   1   1   1
STSite  STSite  Stool   Anterior_nares  Tongue_dorsum   Buccal_mucosa   Supragingival_plaque    Posterior_fornix
Parent_Specimen Parent_Specimen 700023113   700023118   700023124   700023126   700023130   700023121
Run ID  Run ID  615HE   704TB   7055M   61NTD   61NUV   61PJ9
Lane    Lane    4   1   4   3   2   1
SRS SRS 700023113   700023118   700023124   700023126   700023130   700023148
Mean Quality    Mean Quality        33.16   30.61   33.36   31.56   33.04
Number of Quality Bases Number of Quality Bases     4560518567  5153150134  5036725438  5011615225  6263636993
Percent of Human Reads  Percent of Human Reads      0.8114  0.143   0.8042  0.4233  0.7642
Unique Non-Human Bases  Unique Non-Human Bases      584576790   5016349846  903145208   3182590917  1560462076
InverseSimpson  InverseSimpson  48.7504 56.5429 50.3838 50.3593 51.1594 38.9403 73.7574
Shannon Shannon 4.07138 4.24137 4.08732 4.0865  4.09406 3.89726 4.41906
Pielou  Pielou  0.806237    0.839899    0.809393    0.809231    0.810728    0.771756    0.875087
Richness    Richness    1   1   1   1   1   1   1
ko00564 ko00564: Glycerophospholipid metabolism 0.00831617  0.00617307  0.00717039  0.0108354   0.00424018  0.0111103   0.0094699
ko00680 ko00680: Methane metabolism 0.005228    0.00531585  0.00634202  0.00738571  0.00593295  0.00832855  0.00440135
ko00562 ko00562: Inositol phosphate metabolism  0   0.00775703  0   0   0.00272665  0   0.0027541
ko00563 ko00563: Glycosylphosphatidylinositol(GPI)-anchor biosynthesis  0   0.00125469  0   0   0   0   0
ko00561 ko00561: Glycerolipid metabolism    0   0.00471532  0.00589457  0.00893227  0.0020614   0   0.00712117
ko00440 ko00440: Phosphonate and phosphinate metabolism 0.00770176  0.00483651  0   0.00568723  0   0   0.00255154
ko00250 ko00250: Alanine, aspartate and glutamate metabolism    0.0378238   0.0284816   0   0.0170703   0   0.0260858   0.0180229
ko04111 ko04111: Cell cycle - yeast 0   0.000599684 0   0   0   0.00113323  0
ko00010 ko00010: Glycolysis / Gluconeogenesis   0.0137244   0.0209961   0.018308    0.0220677   0.020632    0.0204576   0.0128962
ko00760 ko00760: Nicotinate and nicotinamide metabolism 0.0167863   0.0117542   0.0105919   0.00818106  0.0153611   0.0148305   0.0134239
ko00920 ko00920: Sulfur metabolism  0   0.0119364   0.0130321   0.00589308  0.0142329   0.00856381  0.0101509
ko00311 ko00311: Penicillin and cephalosporin biosynthesis  0   0.00303883  0   0.00272092  0.00175161  0   0
ko00310 ko00310: Lysine degradation 0.0032967   0   0   0   0.00405376  0   0.0022305
ko04146 ko04146: Peroxisome 0.000650666 0.00229658  0   0.00132368  0.00265021  0   0.00246929
ko00600 ko00600: Sphingolipid metabolism    0   0   0   0.00178274  0   0   0.00038175
ko04140 ko04140: Regulation of autophagy    0   0.00305422  0   0   0   0   0
ko04141 ko04141: Protein processing in endoplasmic reticulum    0.00101032  0.000884195 0.000238741 0   0   0   0.000503211
ko04142 ko04142: Lysosome   0   0.000611374 0   0   0   0   0
ko03040 ko03040: Spliceosome    0.000257386 0.00225732  0   0   0   0.000285442 0
ko00523 ko00523: Polyketide sugar unit biosynthesis 0   0.00304293  0   0   0   0.00264827  0
ko02060 ko02060: Phosphotransferase system (PTS)    0.000866446 0.00716477  0.0107349   0.0151732   0.00796107  0.0209403   0.00884015
ko00513 ko00513: Various types of N-glycan biosynthesis 0   0   0   0   0   0.000798493 0
ko00511 ko00511: Other glycan degradation   0.0131206   0.00904807  0.00684248  0.0167268   0.00558892  0.00759045  0.0068108
ko00510 ko00510: N-Glycan biosynthesis  0.00122586  0.00289994  0   0   0.00101664  0   0.000797723
ko05110 ko05110: Vibrio cholerae infection  0   0   0   0   0   0   2.8505e-08
ko04974 ko04974: Protein digestion and absorption   0.00111167  0.000438233 0.000711393 4.208e-05   0   0.00184205  0
:
```

What they're doing: HUMAnN

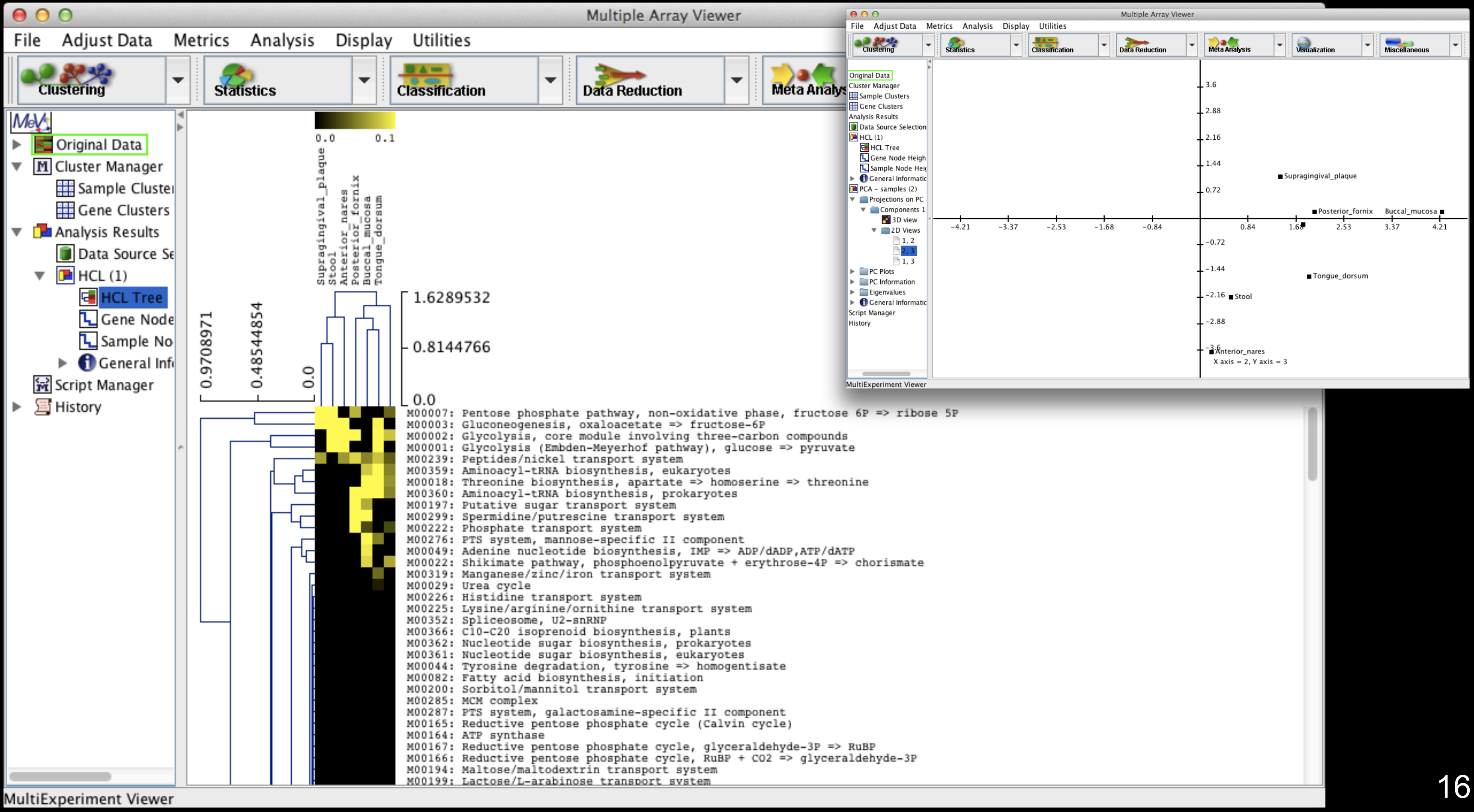
- That's ugly; it gets much better in Excel
  - **Note:** this is very sparse since we're using a small subset of KEGG
  - Note: the mock community demo data is included on the right



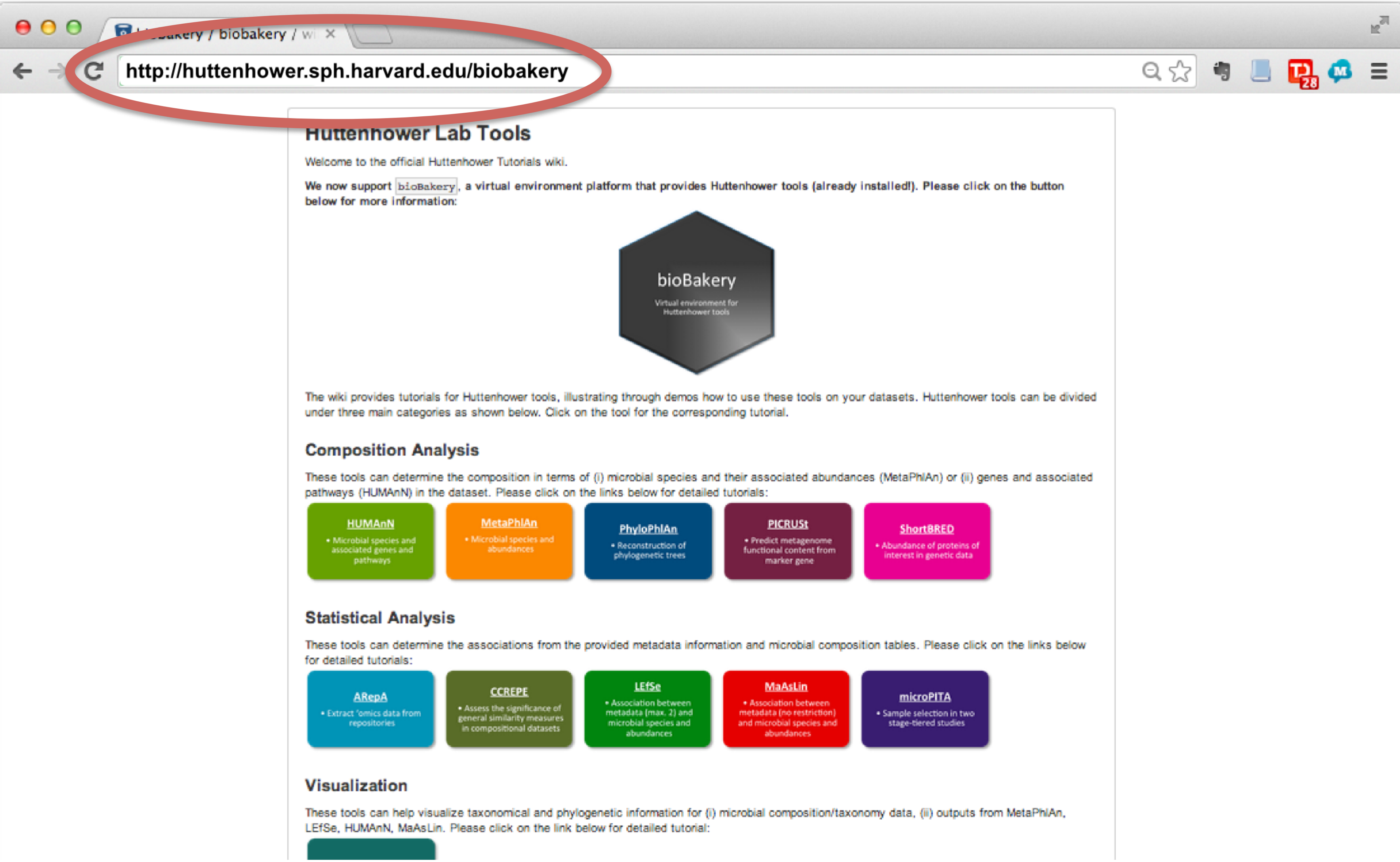
	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	NAME	763577454-SR	763577454-SR	763577454-SR	763577454-SR	763577454-SR	763577454-SR	mock_even_lc-hit-keg-mpm-cop-nul-nve-nve			
2	RANDSID	RANDSID	763577454	763577454	763577454	763577454	763577454	763577454				
3	START	START	Q1_2009	Q1_2009	Q1_2009	Q1_2009	Q1_2009	Q1_2009				
4	GENDER	GENDER	female	female	female	female	female	female				
5	VISNO	VISNO	1	1	1	1	1	1				
6	STSite	STSite	Stool	Anterior_nare:	Tongue_dorsu	Buccal_mucos	Supragingival_	Posterior_forn				
7	Parent_Specin	Parent_Specin	700023113	700023118	700023124	700023126	700023130	700023121				
8	Run ID	Run ID	615HE	704TB	7055M	61NTD	61NUV	61PJ9				
9	Lane	Lane	4	1	4	3	2	1				
10	SRS	SRS	700023113	700023118	700023124	700023126	700023130	700023148				
11	Mean Quality	Mean Quality		33.16	30.61	33.36	31.56	33.04				
12	Number of Qu	Number of Qu		4560518567	5153150134	5036725438	5011615225	6263636993				
13	Percent of Hur	Percent of Hur		0.8114	0.143	0.8042	0.4233	0.7642				
14	Unique Non-H	Unique Non-H		584576790	5016349846	903145208	3182590917	1560462076				
15	InverseSimpso	InverseSimpso	3.83578	6.56415	8.05676	9.43868	2.27165	5.73358	20.2727			
16	Shannon	Shannon	1.36626	1.96258	2.16125	2.31147	0.891905	1.84315	3.11746			
17	Pielou	Pielou	0.268222	0.385291	0.424295	0.453786	0.175098	0.361846	0.612017			
18	Richness	Richness	1	1	1	1	1	1	1			
19	M00171	M00171: C4-d	0	0	0	0	0	0	0			
20	M00173	M00173: Redu	0	0	0	0	0	0	0			
21	M00172	M00172: C4-d	0	0	0	0	0	0	0			
22	M00174	M00174: Meth	0	0	0	0	0	0	0.000488385			
23	M00177	M00177: Ribo:	0	0	0	0	0	0	0			
24	M00176	M00176: Sulfu	0	0	0	0	0	0	0			
25	M00178	M00178: Ribo:	0	0	0	0	0	0	0			
26	M00079	M00079: Kerat	0	0	0	0	0	0	0			
27	M00202	M00202: Oligc	0	0	0	0	0	0	0			
28	M00072	M00072: Oligc	0	0	0	0	0	0	0			
29	M00076	M00076: Derm	0	0	0	0	0	0	0			
30	M00276	M00276: PTS s	0	0	0.0524723	0.125698	0	0	0.00161769			
31	M00229	M00229: Argir	0	0	0	0	0	0	2.96E-05			
32	M00377	M00377: Redu	0	0	0	0	0	0	0			

What they’re doing: HUMAnN

- And there’s nothing stopping us from using MeV
  - Or R, or QIIME, or LEfSe, or anything that’ll read tab-delimited text



```
M00007: Pentose phosphate pathway, non-oxidative phase, fructose 6P => ribose 5P
M00003: Gluconeogenesis, oxaloacetate => fructose-6P
M00002: Glycolysis, core module involving three-carbon compounds
M00001: Glycolysis (Embden-Meyerhof pathway), glucose => pyruvate
M00239: Peptides/nickel transport system
M00359: Aminoacyl-tRNA biosynthesis, eukaryotes
M00018: Threonine biosynthesis, apartate => homoserine => threonine
M00360: Aminoacyl-tRNA biosynthesis, prokaryotes
M00197: Putative sugar transport system
M00299: Spermidine/putrescine transport system
M00222: Phosphate transport system
M00276: PTS system, mannose-specific II component
M00049: Adenine nucleotide biosynthesis, IMP => ADP/dADP,ATP/dATP
M00022: Shikimate pathway, phosphoenolpyruvate + erythrose-4P => chorismate
M00319: Manganese/zinc/iron transport system
M00029: Urea cycle
M00226: Histidine transport system
M00225: Lysine/arginine/ornithine transport system
M00352: Spliceosome, U2-snRNP
M00366: C10-C20 isoprenoid biosynthesis, plants
M00362: Nucleotide sugar biosynthesis, prokaryotes
M00361: Nucleotide sugar biosynthesis, eukaryotes
M00044: Tyrosine degradation, tyrosine => homogentisate
M00082: Fatty acid biosynthesis, initiation
M00200: Sorbitol/mannitol transport system
M00285: MCM complex
M00287: PTS system, galactosamine-specific II component
M00165: Reductive pentose phosphate cycle (Calvin cycle)
M00164: ATP synthase
M00167: Reductive pentose phosphate cycle, glyceraldehyde-3P => RuBP
M00166: Reductive pentose phosphate cycle, RuBP + CO2 => glyceraldehyde-3P
M00194: Maltose/maltodextrin transport system
M00199: Lactose/L-arabinose transport system
```



http://huttenhower.sph.harvard.edu/biobakery

Huttenhower Lab Tools

Welcome to the official Huttenhower Tutorials wiki.

We now support `bioBakery`, a virtual environment platform that provides Huttenhower tools (already installed!). Please click on the button below for more information:

The wiki provides tutorials for Huttenhower tools, illustrating through demos how to use these tools on your datasets. Huttenhower tools can be divided under three main categories as shown below. Click on the tool for the corresponding tutorial.

Composition Analysis

These tools can determine the composition in terms of (i) microbial species and their associated abundances (MetaPhlAn) or (ii) genes and associated pathways (HUMAnN) in the dataset. Please click on the links below for detailed tutorials:

HUMAnN
- Microbial species and associated genes and pathways

MetaPhlAn
- Microbial species and abundances

PhyloPhlAn
- Reconstruction of phylogenetic trees

PICRUSt
- Predict metagenome functional content from marker gene

ShortBRED
- Abundance of proteins of interest in genetic data

Statistical Analysis

These tools can determine the associations from the provided metadata information and microbial composition tables. Please click on the links below for detailed tutorials:

ARepA
- Extract 'omics data from repositories

CCREPE
- Assess the significance of general similarity measures in compositional datasets

LEfSe
- Association between metadata (max. 2) and microbial species and abundances

MaAsLin
- Association between metadata (no restriction) and microbial species and abundances

microPITA
- Sample selection in two stage-tiered studies

Visualization

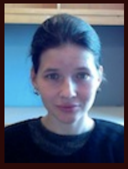
These tools can help visualize taxonomical and phylogenetic information for (i) microbial composition/taxonomy data, (ii) outputs from MetaPhlAn, LEfSe, HUMAnN, MaAsLin. Please click on the link below for detailed tutorial:

Thanks!

http://huttenhower.sph.harvard.edu


Alex Kostic


Levi Waldron


Xochitl Morgan

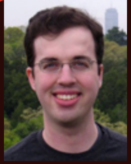

Tim Tickle


Daniela Boernigen

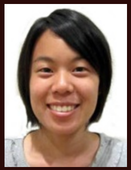

Lauren McIver

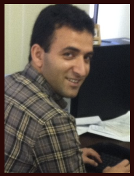

George Weingart


Emma Schwager


Eric Franzosa


Boyu Ren


Tiffany Hsu


Ali Rahnavard


Ayshwarya Subramanian


Jim Kaminski


Regina Joice


Koji Yasuda


Kevin Oh


Galeb Abu-Ali


Afrah Shafquat


Randall Schwager


Chengwei Luo


Keith Bayer


Moran Yassour


Alexandra Sirota


Dirk Gevers


Ramnik Xavier

Human Microbiome Project 2

Lita Procter
Jon Braun
Dermot McGovern
Subra Kugathasan
Ted Denson
Janet Jansson

Bruce Birren
Chad Nusbaum
Clary Clish
Joe Petrosino
Thad Stappenbeck

Human Microbiome Project

Jane Peterson
Sarah Highlander
Barbara Methe

Karen Nelson
George Weinstock
Owen White


Wendy Garrett


Nicola Segata


Gautam Dantas
Molly Gibson


Andy Chan


Katherine Lemon


Brendan Bohannan
James Meadow