# Metagenomika – Vyhodnocování dat

Petra Vídeňská Ph.D.

# NGS formáty

- Samotná sekvence + informace o kvalitě jednotlivých nukleotidů
- 454 → .sff – lze rozdělít na dva podsoubory .fasta a .qual (kvalita)
- Illumina, IonTorrent - FastQ

# FastQ formát

```
@HWUSI-EAS582_157:6:1:1:1501/1
NCACAGACACACACGAACACACAAAGACATGCCCATATGAAGAT
+
%.7786867:7785568587465750588 73/347777476035
@HWUSI-EAS582_157:6:1:1:1606/1
NCTGGCACCTTGATTTTGGACTTCCCAGCCTCCAGAACTGTGAG
+
%1948988888798988366898888648998788898888588
```

- Line 1 : begins with @ plus sequence identifier
- Line 2 : sequence(“read”)
- Line 3 : begins +
- Line 4 : quality values

# FastQ formát

*@HWUSI-EAS100R:1:2:99:88#0/1*

- 1: flowcell number
- 2: tile number
- 99: “x” - coordinate
- 88: “y” - coordinate
- #0: number of index

# FastQ formát – quality score

| Q | P | Base was call right |
|---|---|---|
| 10 | 0.1 | 90% (1 from 10) |
| 20 | 0.01 | 99% (1 from 100) |
| 30 | 0.001 | 99.9% (1 from 1000) |
| 40 | 0.0001 | 99.99% (1 from10 000) |

- $Q = -10 \log_{10} P$

*P…probability base call is incorrect*

# Fast QC

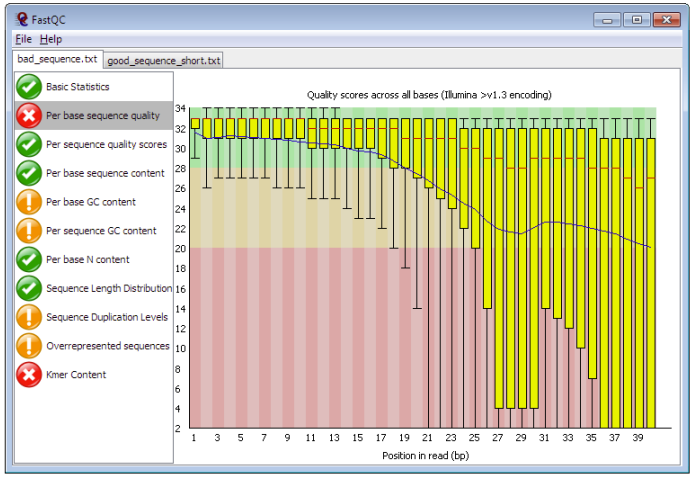
www.bioinformatics.babraham.ac.uk/projects/fastqc/

About | People | Services | Projects | Training | Publications

## FastQC

| | |
|---|---|
| **Function** | A quality control tool for high throughput sequence data. |
| **Language** | Java |
| **Requirements** | A suitable Java Runtime Environment<br>The Picard BAM/SAM Libraries (included in download) |
| **Code Maturity** | Stable. Mature code, but feedback is appreciated. |
| **Code Released** | Yes, under GPL v3 or later. |
| **Initial Contact** | Simon Andrews |

**Download Now**

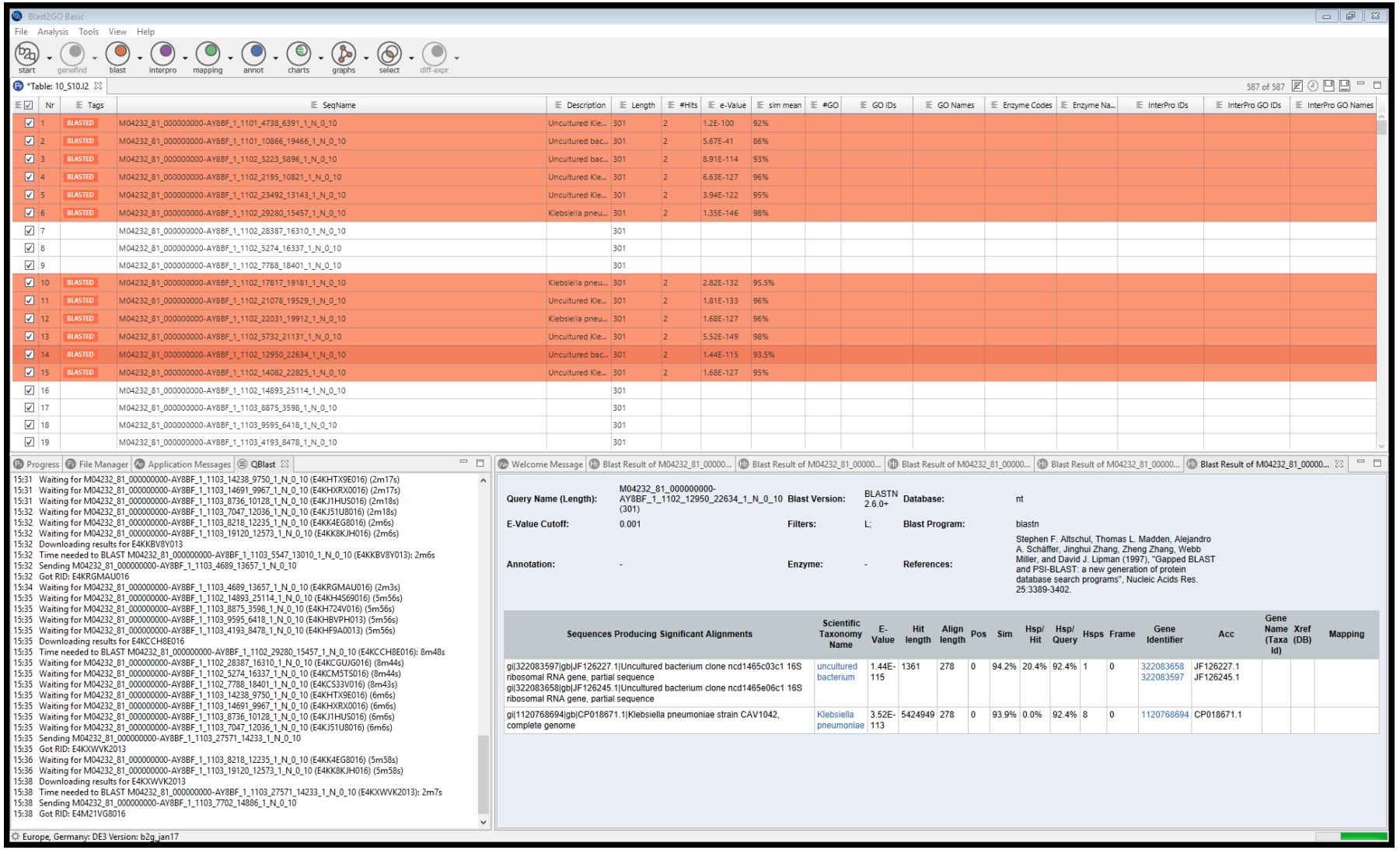



View our tutorial video

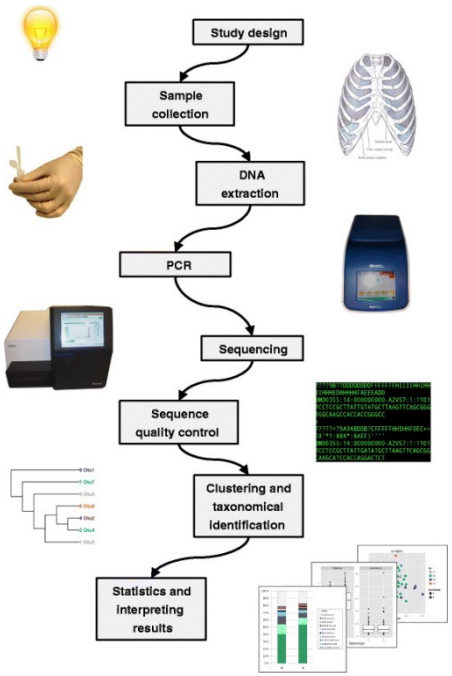
FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a modular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

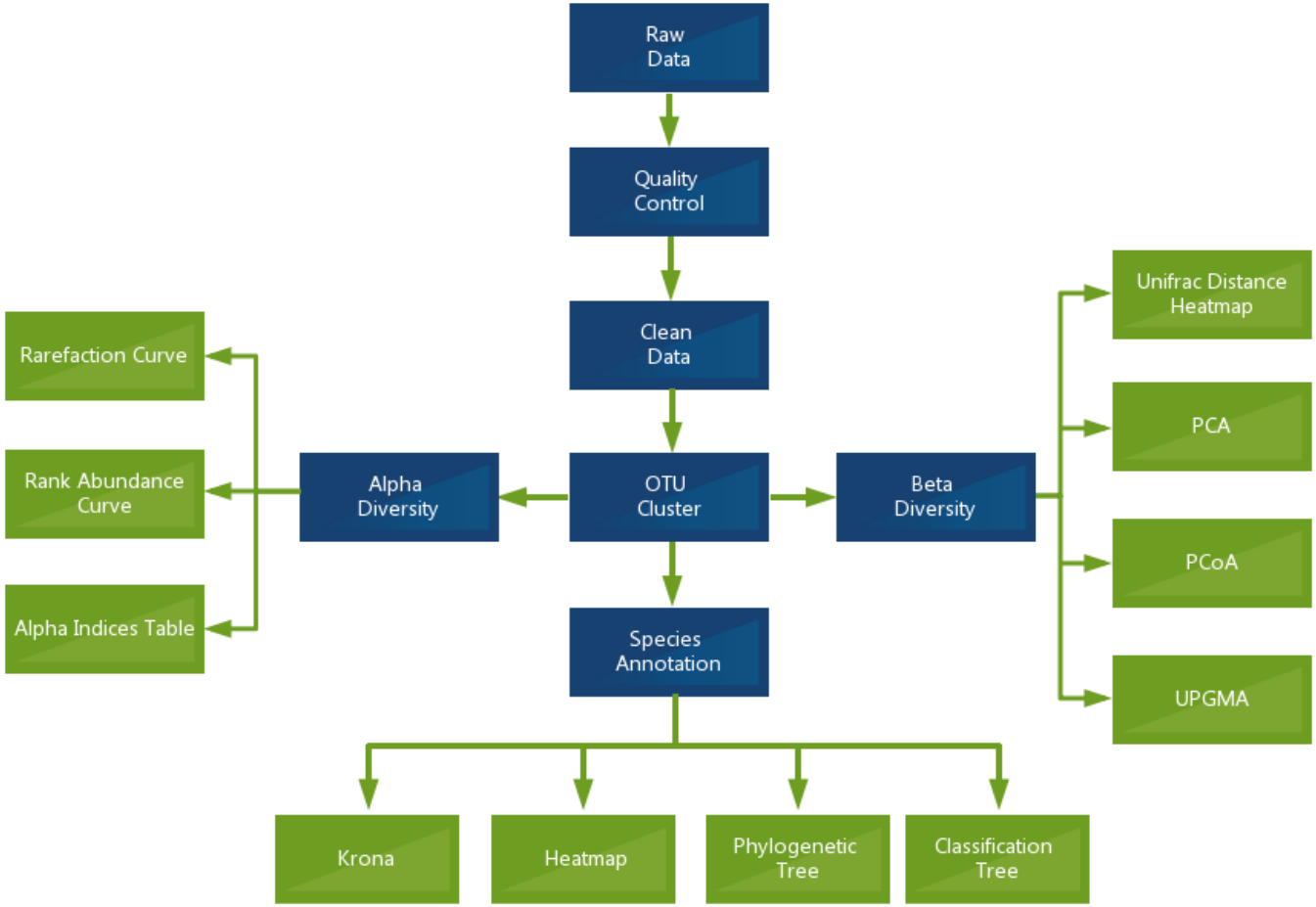
The main functions of FastQC are

# Blast2go


Blast2GO Basic
File Analysis Tools View Help
start geneInd blast interpro mapping annot charts graphs select diff-expr
*Table: 10_S10.l2
587 of 587
Nr Tags SeqName Description Length #Hits e-Value sim mean #GO GO IDs GO Names Enzyme Codes Enzyme Na... InterPro IDs InterPro GO IDs InterPro GO Names
1 BLASTED M04232_81_000000000-AY8BF_1_1101_4738_6391_1_N_0_10 Uncultured Kle... 301 2 1.2E-100 92%
2 BLASTED M04232_81_000000000-AY8BF_1_1101_10866_19466_1_N_0_10 Uncultured bac... 301 2 5.67E-41 86%
3 BLASTED M04232_81_000000000-AY8BF_1_1102_5223_5896_1_N_0_10 Uncultured bac... 301 2 8.91E-114 93%
4 BLASTED M04232_81_000000000-AY8BF_1_1102_2195_10821_1_N_0_10 Uncultured Kle... 301 2 6.63E-127 96%
5 BLASTED M04232_81_000000000-AY8BF_1_1102_23492_13143_1_N_0_10 Uncultured Kle... 301 2 3.94E-122 95%
6 BLASTED M04232_81_000000000-AY8BF_1_1102_29280_15457_1_N_0_10 Klebsiella pneu... 301 2 1.35E-146 98%
7 M04232_81_000000000-AY8BF_1_1102_28387_16310_1_N_0_10 301
8 M04232_81_000000000-AY8BF_1_1102_5274_16337_1_N_0_10 301
9 M04232_81_000000000-AY8BF_1_1102_7788_18401_1_N_0_10 301
10 BLASTED M04232_81_000000000-AY8BF_1_1102_17817_19181_1_N_0_10 Klebsiella pneu... 301 2 2.82E-132 95.5%
11 BLASTED M04232_81_000000000-AY8BF_1_1102_21078_19529_1_N_0_10 Uncultured Kle... 301 2 1.81E-133 96%
12 BLASTED M04232_81_000000000-AY8BF_1_1102_22031_19912_1_N_0_10 Klebsiella pneu... 301 2 1.68E-127 96%
13 BLASTED M04232_81_000000000-AY8BF_1_1102_5732_21131_1_N_0_10 Uncultured Kle... 301 2 5.52E-149 98%
14 BLASTED M04232_81_000000000-AY8BF_1_1102_12950_22634_1_N_0_10 Uncultured bac... 301 2 1.44E-115 93.5%
15 BLASTED M04232_81_000000000-AY8BF_1_1102_14082_22825_1_N_0_10 Uncultured Kle... 301 2 1.68E-127 95%
16 M04232_81_000000000-AY8BF_1_1102_14893_25114_1_N_0_10 301
17 M04232_81_000000000-AY8BF_1_1103_8875_3598_1_N_0_10 301
18 M04232_81_000000000-AY8BF_1_1103_9595_6418_1_N_0_10 301
19 M04232_81_000000000-AY8BF_1_1103_4193_8478_1_N_0_10 301
Progress File Manager Application Messages QBlast
15:31 Waiting for M04232_81_000000000-AY8BF_1_1103_14238_9750_1_N_0_10 (E4KHTX9E016) (2m17s)
15:31 Waiting for M04232_81_000000000-AY8BF_1_1103_14691_9967_1_N_0_10 (E4KHXRX0016) (2m17s)
15:31 Waiting for M04232_81_000000000-AY8BF_1_1103_8736_10128_1_N_0_10 (E4KJ1HUS016) (2m18s)
15:32 Waiting for M04232_81_000000000-AY8BF_1_1103_7047_12036_1_N_0_10 (E4KJ51U8016) (2m18s)
15:32 Waiting for M04232_81_000000000-AY8BF_1_1103_8218_12235_1_N_0_10 (E4KK4EG8016) (2m6s)
15:32 Waiting for M04232_81_000000000-AY8BF_1_1103_19120_12573_1_N_0_10 (E4KK8KJH016) (2m6s)
15:32 Downloading results for E4KKBV8Y013
15:32 Time needed to BLAST M04232_81_000000000-AY8BF_1_1103_5547_13010_1_N_0_10 (E4KKBV8Y013): 2m6s
15:32 Sending M04232_81_000000000-AY8BF_1_1103_4689_13657_1_N_0_10
15:32 Got RID: E4KRGMAU016
15:34 Waiting for M04232_81_000000000-AY8BF_1_1103_4689_13657_1_N_0_10 (E4KRGMAU016) (2m3s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1102_14893_25114_1_N_0_10 (E4KH4S69016) (5m56s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_8875_3598_1_N_0_10 (E4KH724V016) (5m56s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_9595_6418_1_N_0_10 (E4KHBVPH013) (5m56s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_4193_8478_1_N_0_10 (E4KHF9A0013) (5m56s)
15:35 Downloading results for E4KCCH8E016
15:35 Time needed to BLAST M04232_81_000000000-AY8BF_1_1102_29280_15457_1_N_0_10 (E4KCCH8E016): 8m48s
15:35 Waiting for M04232_81_000000000-AY8BF_1_1102_28387_16310_1_N_0_10 (E4KCGUJG016) (8m44s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1102_5274_16337_1_N_0_10 (E4KCM5TS016) (8m44s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1102_7788_18401_1_N_0_10 (E4KCS33V016) (8m43s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_14238_9750_1_N_0_10 (E4KHTX9E016) (6m6s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_14691_9967_1_N_0_10 (E4KHXRX0016) (6m6s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_8736_10128_1_N_0_10 (E4KJ1HUS016) (6m6s)
15:35 Waiting for M04232_81_000000000-AY8BF_1_1103_7047_12036_1_N_0_10 (E4KJ51U8016) (6m6s)
15:35 Sending M04232_81_000000000-AY8BF_1_1103_27571_14233_1_N_0_10
15:35 Got RID: E4KXWVK2013
15:36 Waiting for M04232_81_000000000-AY8BF_1_1103_8218_12235_1_N_0_10 (E4KK4EG8016) (5m58s)
15:36 Waiting for M04232_81_000000000-AY8BF_1_1103_19120_12573_1_N_0_10 (E4KK8KJH016) (5m58s)
15:38 Downloading results for E4KXWVK2013
15:38 Time needed to BLAST M04232_81_000000000-AY8BF_1_1103_27571_14233_1_N_0_10 (E4KXWVK2013): 2m7s
15:38 Sending M04232_81_000000000-AY8BF_1_1103_7702_14886_1_N_0_10
15:38 Got RID: E4M21VG8016
Europe, Germany: DE3 Version: b2g_jan17
Welcome Message
Blast Result of M04232_81_00000...
Blast Result of M04232_81_00000...
Blast Result of M04232_81_00000...
Blast Result of M04232_81_00000...
Blast Result of M04232_81_00000...
Query Name (Length): M04232_81_000000000-AY8BF_1_1102_12950_22634_1_N_0_10 (301)
Blast Version: BLASTN 2.6.0+
Database: nt
E-Value Cutoff: 0.001
Filters: L;
Blast Program: blastn
Annotation: -
Enzyme: -
References: Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.
Sequences Producing Significant Alignments | Scientific Taxonomy Name | E-Value | Hit length | Align length | Pos | Sim | Hsp/ Hit | Hsp/ Query | Hsps | Frame | Gene Identifier | Acc | Gene Name (Taxa Id) | Xref (DB) | Mapping
gi|322083597|gb|JF126227.1|Uncultured bacterium clone ncd1465c03c1 16S ribosomal RNA gene, partial sequence gi|322083658|gb|JF126245.1|Uncultured bacterium clone ncd1465e06c1 16S ribosomal RNA gene, partial sequence | uncultured bacterium | 1.44E-115 | 1361 | 278 | 0 | 94.2% | 20.4% | 92.4% | 1 | 0 | 322083658 322083597 | JF126227.1 JF126245.1
gi|1120768694|gb|CP018671.1|Klebsiella pneumoniae strain CAV1042, complete genome | Klebsiella pneumoniae | 3.52E-113 | 5424949 | 278 | 0 | 93.9% | 0.0% | 92.4% | 8 | 0 | 1120768694 | CP018671.1

# Analýza dat genu pro 16S rRNA - workflow



# Úprava sekvencí - délka, kvalita

```
>0 17_6006268
GGCCCTCGGGTTGTAAACTTCGTTTTATCAGGGACGAAGCAAGTGACGGTACCTGATGAATAAGCCACGGCTA
ACTACGTGCCAGCAGCCGCGGTAATACGTAGGTGGCGAGCGTTATCCGGATTTACTGGGTGTAAAGGGCGTGT
AGGCGGGAAGGCAAGTCAGACGTGAAAACCACGGGCTCAACCTGTGGCCTGCATTTGAAACTGTAGTTCTTGA
GTGCTGGAGAGGCAATCGGAATTCCGTGTGTAGCGGTGAAATGCGTAGATATACGGAGGAACACCAGTGGCGA
AGGCGGATTGCTGGACGATAACTGACGCTGAGGCGCGAAAGCGTGGGG
>1 8_4033127
GTGCTTCAGCGTCCGTCCTGGCCTGGTATGCTGCCTTCGCAATCGGGGTTCTGCGTGATATCTATGCATTTCA
CCGCTACACCATACATTCCGCCTACCGCAACTACTCTCTAGCCCAACAGTATTGGAGGCAGTTTCAGGGTTAA
GCCCTAACATTTCACCTCCAACTTATCGAACCGCCTACGCACCCTTTAAACCCAATAAATCCGGATAACGCTT
GGATCCTCCGTATTACCGCGGCTGCTGGCACGGAGTTAGCCGATCCTTACTCGTACGATACTTTCAGACAGAT
ACACGTATCTGCGTTTACCCTCGTACAAAAGCAGTTTACAACTCATAG
>10 14_1017803
CTGCCTTCGCGATCGGAGTTCTTCGTGATATCTAAGCATTTCACCGCTACACTCGGAATTCCGCCAACCTCTA
CTTCACTCAAGAAAACCAGTTTCAACTGCAGTCTACAGGTTAAGCCCGTAGTTTTCACAGCTGACTTGGCTCC
CCGCCTGCGCTCCCTTTACACCCAGTAATTCCGGACAACGCTTGCCACCTACGTATTACCGCGGCTGCTGGCA
CGTAGTTAGCCGTGGCTTGCTCCTTAGCTACCGTCACTATCTTCACTAAGAACAGAAGTTTACAATCCGAAAA
CCGTCTTCCTTCACGCGGCGTTGCTGCATCAGGGTTTCCCCCATTGTG
>100 17_6006779
AAGGCGTTTCGGGTTGTAAACTTCTTTTCTCAGGGACGAAGAAAATGACGGTACCTGAGGAATAAGCCACGGC
TAACTACGTGCCAGCAGCCGCGGTAATACGTAGGTGGCAAGCGTTGTCCGGATTTACTGGGTGTAAAGGGCGC
GTAGGCGGGGAGACAAGTCAGATGTGAAAACCAGGGGCTCAACCTCTGGCCTGCATTTGAAACTGTAGCTCTT
GAGTGTCGGAGAGGCAATCGGAATTCCGTGTGTAGCGGTGAAATGCGTAGATATACGGAGGAACACCAGTGGC
GAAGGCGGATTGCTGGACGATAACTGACGCTGAGGCGCGAAAGCGTGG
>1000 15_1026250
CTGCCTTCGCAATCGGGGTTCTTCGTGATATCTATGCATTTCACCGCTACACCACGAATTCCGCATGCCGCGA
CCGTACTCAAGCCCCACAGTTTCAACTGCAATTTTACGGTTGAGCCGCAAACTTTCACAGCTGACTTAAGGGG
CCGTCTGCGCTCCCTTTAAACCCAATAAACTCCGGATAACGCTCGCATCCTCCGTATTACCGCGGCTGCTGGC
ACGGAGTTAGCCGATGCTTTTTCTTCGGATACTTGCAATACGCTACACGTAGCGCACTTTACTCTCCGACAAA
ACGAAGTTTACAACCCGTAGGGCCGTCTTCCTTCACGCGACTTGGCTG
>10000 K3.1_20000229
GTATCTCGGTATGTAAAGCTCTATCAGCAGGGAAGAAAATGACGGTACCTGACTAAGAAGCACCGGCTAAATA
CGTGCCAGCAGCCGCGGTAATACGTATGGTGCAAGCGTTATCCGGATTTACTGGGTGTAAAGGGAGCGTAGAC
GGAGAAGCAAGTCTGGAGTGAAAACCCGGGGCTCAACCCCGGGACTGCTTTGGAAACTGTTTTTCTAGAGTGC
CGGAGAGGTAAGCGGAATTCCTAGTGTAGCGGTGAAATGCGTAGATATTAGGAGGAACACCAGTGGCGAAGGC
GGCTTACTGGACGGTAACTGACGTTGAGGCTCGAAAGCGTGGGAGCAA
>10001 14_1025931
```

- Filtrace příliš krátkých sekvencí
- Filtrace nekvalitních sekvencí
- Trimmování sekvencí na základě kvality (odstranění nekvalitních bází)

# Spojení sekvencí a rozřazení na základě značek

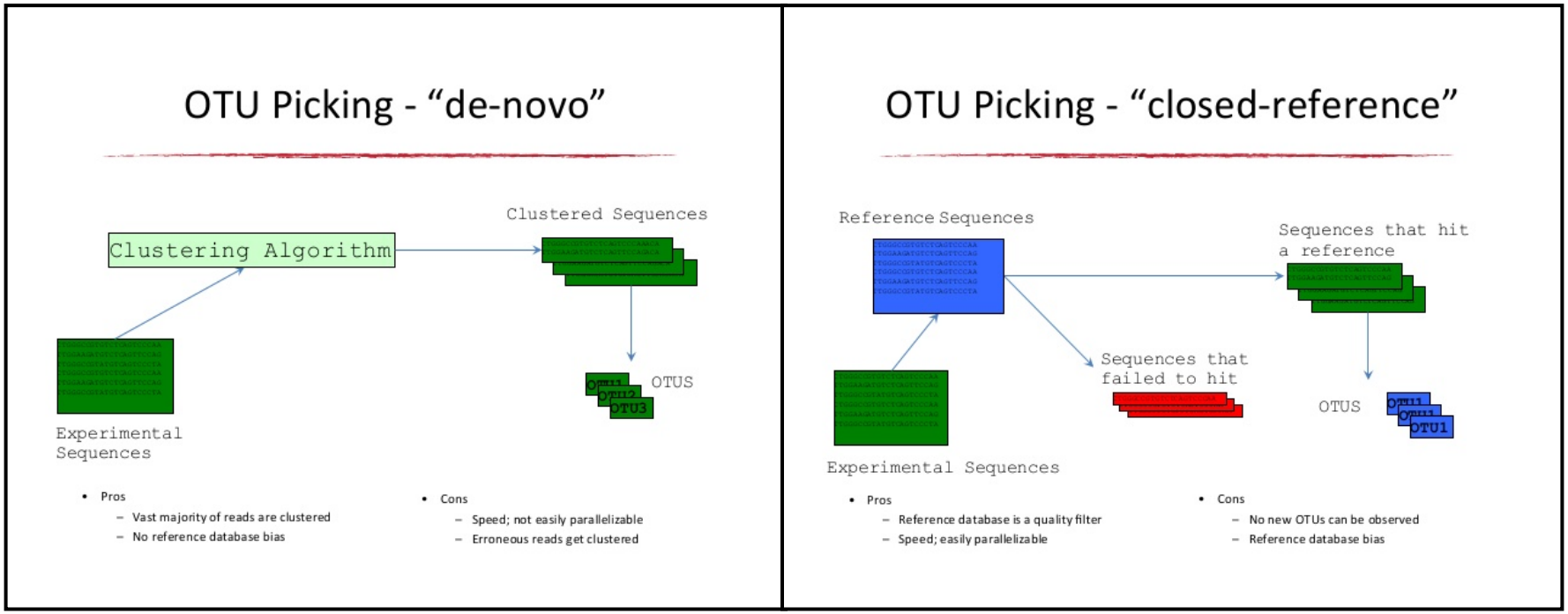
```
>16_5000037 HN7PVAR01BC90H orig_bc=TAGTATCAGC new_bc=TAGTATCAGC
bc_diffs=0
TGTTTGCTACCCATGCTTTCGAGCCTCAGCGTCAGTTAGTGCCCAGTAGGCCGCCTTCGCCACTGGTGTTCCT
CCCGATATCTACGCATTCCACCGCTACACCGGGAATTCCGCCTACCTCTACACCACTCAAGATCCACAGTTTT
GAAAGCAGTTCATGGGTTGAGCCCATGTATTTCACTTCCAACTTGCATACCCGCCTGCGCTCCCTTTACACCC
AGTAATTCCGGACAACGCTTGTGACCTACGTTTTACCGCGGCTGCTGGCACGTAGTTAGCCGTCACTTCCTTG
TTGGGTACCATCCGTATTTTCCCCAACAACAGGAGTTTACAATCCGAAGACCTTCTTCCTCCACGCGGCGTCG
CTGCATCAGGGTTTCCCCCATTGTGCAAT
>17_5000038 HN7PVAR01ALS7Q orig_bc=CGTGTCTCTA new_bc=CGTGTCTCTA
bc_diffs=0
TGTTTGCTCCCCACGCTTTCGCGCCTCACCGTCAGTTGCCGTCCAGTTATCCGCCTTCGCCACTGGTGTTCTT
CCTTATATCTACGCATTTCACCGCTACACAAGGAATTCCGATAACCTCTCCGGTACTCAAGACCAGCAGTTTC
AAATGCAGTTCGTGGGTTAAGCCCACGCATTTCACATCTGACTTGCCAGCCCGGCTGCACGCCCTTTACACCC
AGTAAATCCGGACAACGCTTGCCACCTACGTATTACCGCGGCTGCTGGCACGTAGTTAGCCGTGGCTTATTCG
TCAGGTACCCTCTTCTACTGTTCCCTGACAAAAGAAGTTTACAACCCGAGGGCCTTCTTCCTTCACGCGGCGT
TGCTGGGTCAGGCTTGCGCCCATTGCCCAAT
>16_5000039 HN7PVAR01AGDPN orig_bc=TAGTATCAGC new_bc=TAGTATCAGC
bc_diffs=0
TGTTTGATACCCACGCTTTCGAGCTTCAATGTCAGTCGCGGCTTGGTGGACTGCCTGCGCAATCGGAGTTCTT
CGTGATATCTATGCATTTCACCGCTACACCACGAATTCCATCCACCGCAAACGCACTCAAGACTGCCAGTTTC
AACTGCAGGCCGACGGTTGAGCCGCCGGATTTCACAACTGACTTAACAGCCCATCTACGCTCCCTTTAAACCC
AATAAATCCGGATAACGCTCGCATCCTACGTATTACCGCGGCTGCTGGCACGTAGTTAGCCGTGGCTTCCTCG
TACACTACCGTCATTACCAGCCATTATTCACAACCGGCACATTCGTCATGTACAACAGAGCTTTACAATCCGA
AGACCTTCTTCACTCACGCGGCGTTGCTCCGTCAGGCTTTCGTCCCACTTGCGGAAGATTCCCCTACCTGCTG
CCTCCGCTGATACTA
>16_5000040 HN7PVAR01AVMTO orig_bc=TAGTATCAGC new_bc=TAGTATCAGC
bc_diffs=0
TGTTTGATACCCGCACTTTCGAGCATCAACGTCAGTTACGGTCCAGCAAGCTGCCTTCGCAATCGGAGTTCTT
CGTGATATCTAGGCATTTCACCGCTACACTAGGAATTCCGCCTGCCTCTCCAGTACTCAAGAACTACAGTTTC
AAATGCAGTTCCGGGGTTGAGCCCCGGAATTTCACATCTGACTTGCAATCCCGCCTACACGCCCTTTACACCC
AGTAAATCCGGACAACGCTCGCTCCCTACGTATTACCGCGGCTGCTGGCACGTAGTTAGCCGGAGCTTATTCG
TCAGGTACCCTCTTCTACTGTTCCCTGACAAAAGAAGTTTACAACCCGAAACGCCTTCTTCCTTCACGCGGCG
TTGCTGGGTCAGGCTTGCGCCCATTGCCCAACTACTTCCCCACCTGCTGCCTCCGCTGATACTA
>11_5000041 HN7PVAR01AC3MJ orig_bc=CTCGCGTGTC new_bc=CTCGCGTGTC
bc_diffs=0
TGTTTGATACCCACGCTTTCGAGCTTCAATGTCAGTTGCGGCTTGGTGGACTGCCTTCGCAATCGGAGTTCTT
CGTGATATCTAAGCATTTCACCGCTACACCACGAATTCCATCCACCGCAAACGCACTCAAGGCTACCAGTTTC
AATTGCAGTTCAACGGTTGAGCCGCTGACTTTCACAACTGACTTAACAGCCCATCTACGCTCCCTTTAAACCC
AATAAATCCGGATAACGCTCGCATCCTCCGTATTACCGCGGCTGCTGGCACGGAGTTAGCCGATGCTTATTCG
TACGGTACATGCACACGACCACACGTGGACGCCGTTATTCCCGTACAAAAGGAGTTTACAACCCGTAGGCCGT
CTTCCTCCACGCTACTTGGCTGGTTCAGGCTCACGCCCATTGACCAAT
>7_5000042 HN7PVAR01A8NE0 orig_bc=AGCACTGTAG new_bc=AGCACTGTAG bc_diffs=0
```

- Spojení sekvencí na základě námi zadaných parametrů (min. překryv, počet chyb - mismatch)
- Tvorba mapy, kde uvedeme název vzorku a sekvenci jeho značky
- Rozřazení o označení sekvencí podle vzorku

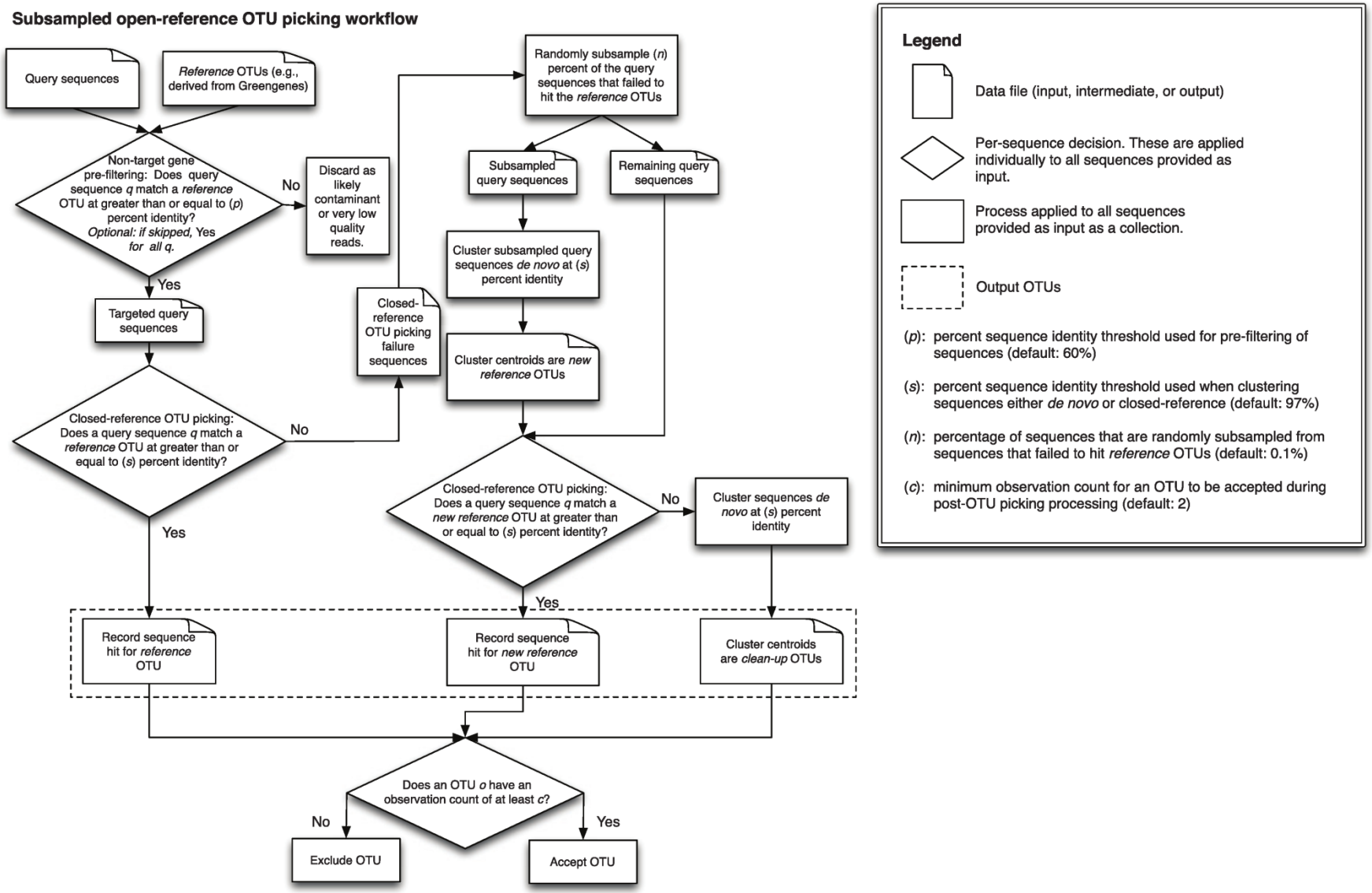
# Ukázka Mapy

| #SampleID | BarcodeSequence | LinkerPrimerSequence | ReversePrimer | Treatment | Description |
|---|---|---|---|---|---|
| 1 | ACGCTCGACA | GGAGGCAGCAGTRRGGAAT | CTACCRGGGTATCTAATCC | Control_1 | stre_control_chicken_0 |
| 2 | AGACGCACTC | GGAGGCAGCAGTRRGGAAT | CTACCRGGGTATCTAATCC | Control_2 | tet_control_chicken_0 |
| 3 | AGCACTGTAG | GGAGGCAGCAGTRRGGAAT | CTACCRGGGTATCTAATCC | atb_1 | stre_chicken_2b_I.D._3 |
| 4 | ATCAGACACG | GGAGGCAGCAGTRRGGAAT | CTACCRGGGTATCTAATCC | atb_2 | tet_chicken_2b_I.D._4 |
| | | | | | |
| #SampleID | BarcodeSequence | LinkerPrimerSequence | ReversePrimer | Treatment | Description |
| 1 | ACGCTCGACA | CTACCRGGGTATCTAATCC | GGAGGCAGCAGTRRGGAAT | Control_1 | stre_control_chicken_0 |
| 2 | AGACGCACTC | CTACCRGGGTATCTAATCC | GGAGGCAGCAGTRRGGAAT | Control_2 | tet_control_chicken_0_I |
| 3 | AGCACTGTAG | CTACCRGGGTATCTAATCC | GGAGGCAGCAGTRRGGAAT | atb_1 | stre_chicken_2b_I.D._3 |
| 4 | ATCAGACACG | CTACCRGGGTATCTAATCC | GGAGGCAGCAGTRRGGAAT | atb_2 | tet_chicken_2b_I.D._4 |

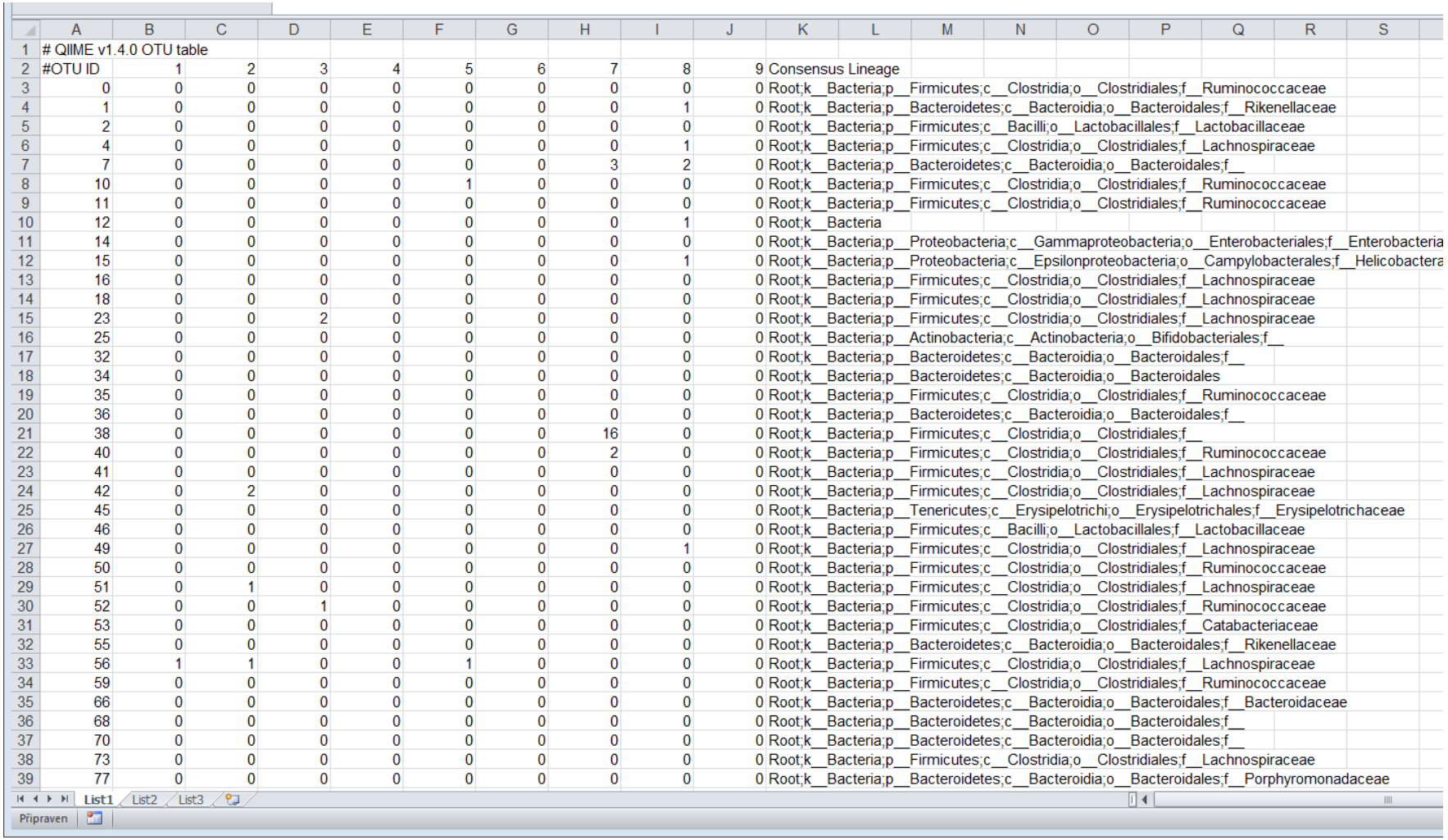
# Tvorba OTUs na základě 97% podobnosti



# Tvorba OTUs na základě 97% podobnosti



# OTU Tabulka



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | # QIIME v1.4.0 OTU table | | | | | | | | | | | | | | | | | | |
| 2 | #OTU ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Consensus Lineage | | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Rikenellaceae | | | | | | | | |
| 5 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Bacilli;o__Lactobacillales;f__Lactobacillaceae | | | | | | | | |
| 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 7 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__ | | | | | | | | |
| 8 | 10 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 9 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 10 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Root;k__Bacteria | | | | | | | | |
| 11 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Proteobacteria;c__Gammaproteobacteria;o__Enterobacteriales;f__Enterobacteria | | | | | | | | |
| 12 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Root;k__Bacteria;p__Proteobacteria;c__Epsilonproteobacteria;o__Campylobacterales;f__Helicobactera | | | | | | | | |
| 13 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 14 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 15 | 23 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 16 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Actinobacteria;c__Actinobacteria;o__Bifidobacteriales;f__ | | | | | | | | |
| 17 | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__ | | | | | | | | |
| 18 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales | | | | | | | | |
| 19 | 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 20 | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__ | | | | | | | | |
| 21 | 38 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__ | | | | | | | | |
| 22 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 23 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 24 | 42 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 25 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Tenericutes;c__Erysipelotrichi;o__Erysipelotrichales;f__Erysipelotrichaceae | | | | | | | | |
| 26 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Bacilli;o__Lactobacillales;f__Lactobacillaceae | | | | | | | | |
| 27 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 28 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 29 | 51 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 30 | 52 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 31 | 53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Catabacteriaceae | | | | | | | | |
| 32 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Rikenellaceae | | | | | | | | |
| 33 | 56 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 34 | 59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae | | | | | | | | |
| 35 | 66 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Bacteroidaceae | | | | | | | | |
| 36 | 68 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__ | | | | | | | | |
| 37 | 70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__ | | | | | | | | |
| 38 | 73 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Lachnospiraceae | | | | | | | | |
| 39 | 77 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Root;k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Porphyromonadaceae | | | | | | | | |

List1 List2 List3

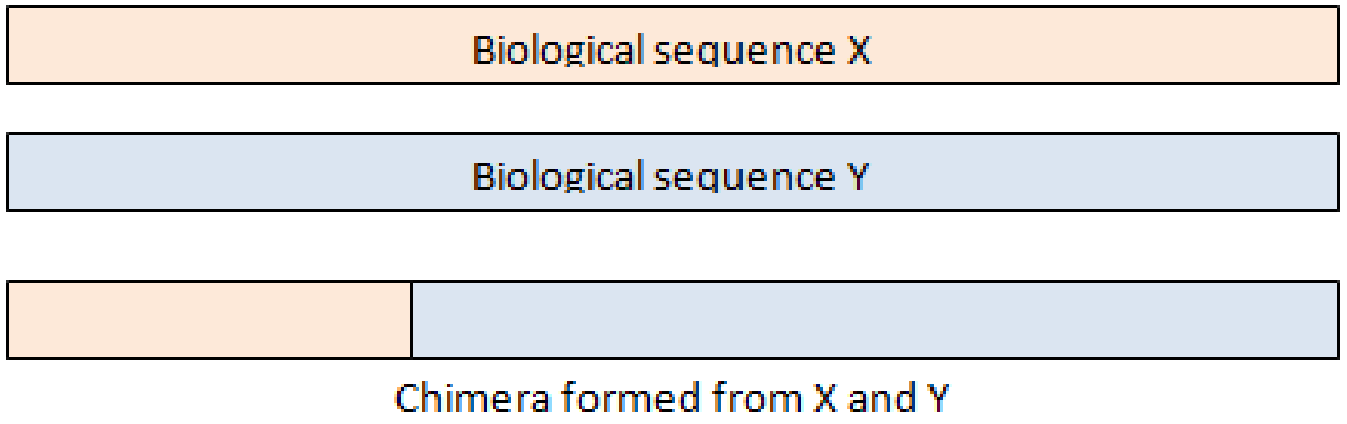
Připraven

# Tvorba reprezentativních sekvencí a jejich taxonomické zařazení (vybraná databáze)

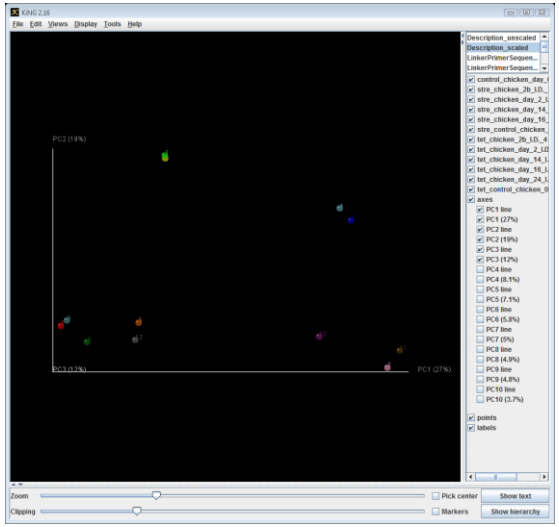
- Referenční sekvence =
  - Nejčetnější sekvence
  - Seed (první) sekvence
  - Konsenzuální sekvence v rámci daného OTU
- Nejznámější databáze
  - RDP
  - SILVA
  - GreenGenes

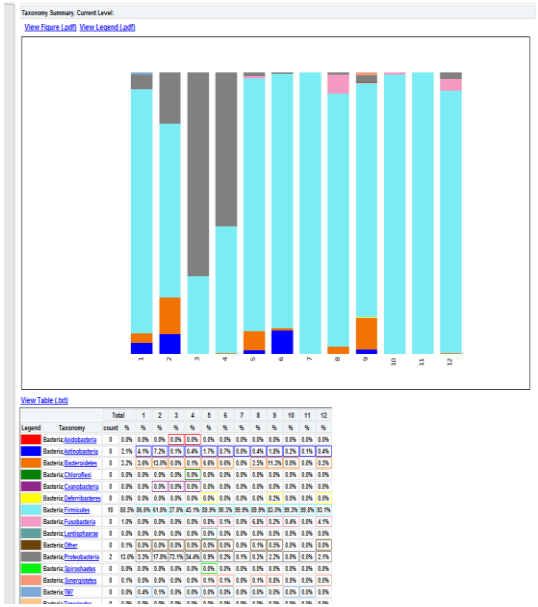
# Odstranění chimér

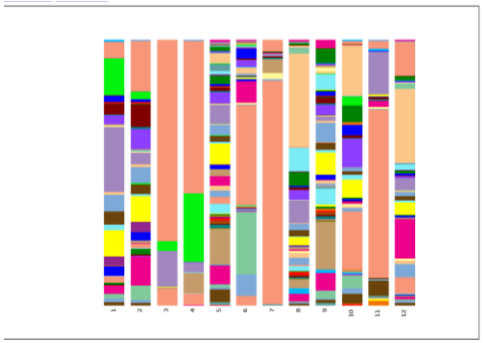
- Pomocí algoritmu odstranění chimér vznikajících při PCR, obvykle 5-20 % sekvencí
- Problém – odlišení od neznámých bakteriálních genomů, může odstraňovat i správné sekvence nebo chiméry neodhalit

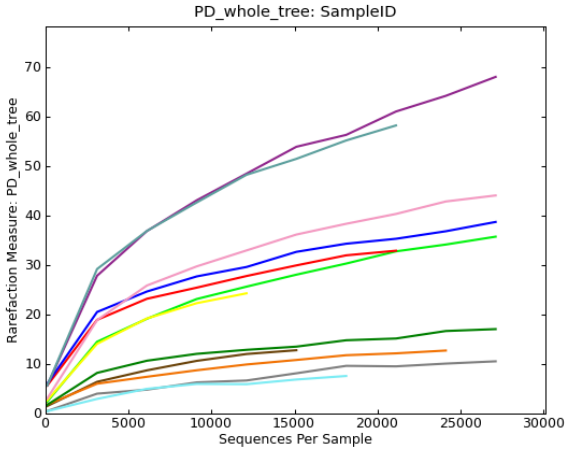


# Ukázka výstupů - grafy

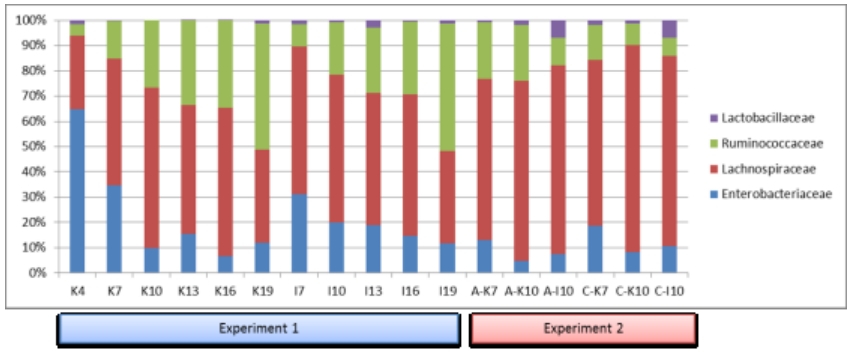


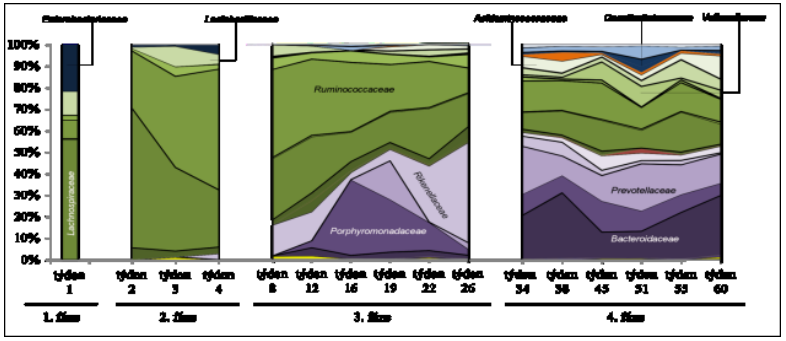


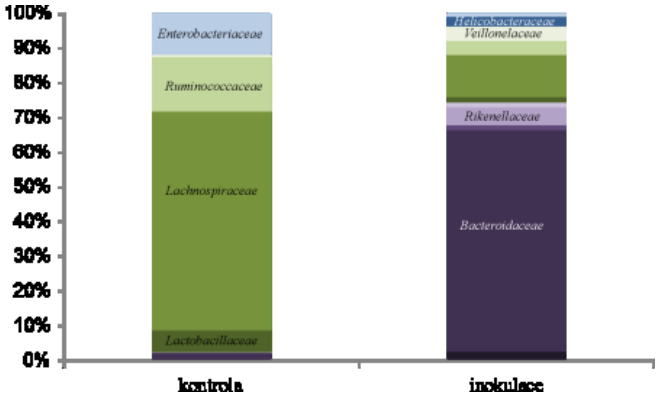


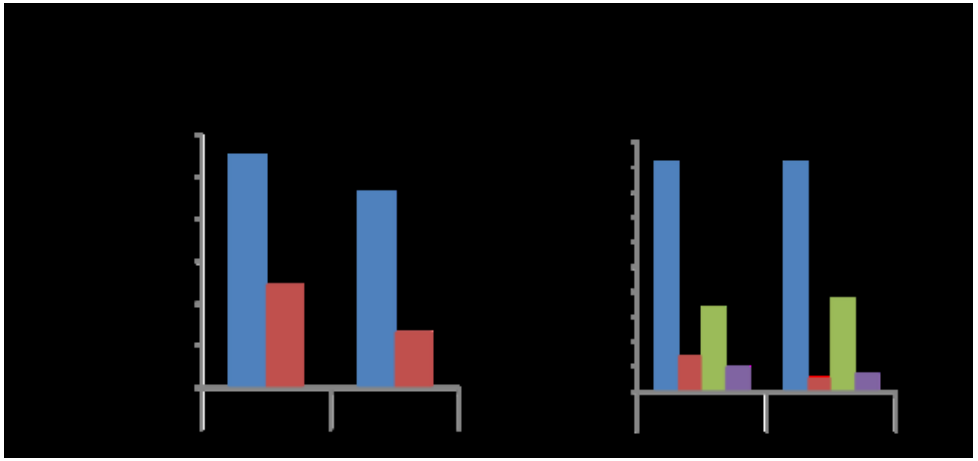


# Analýza

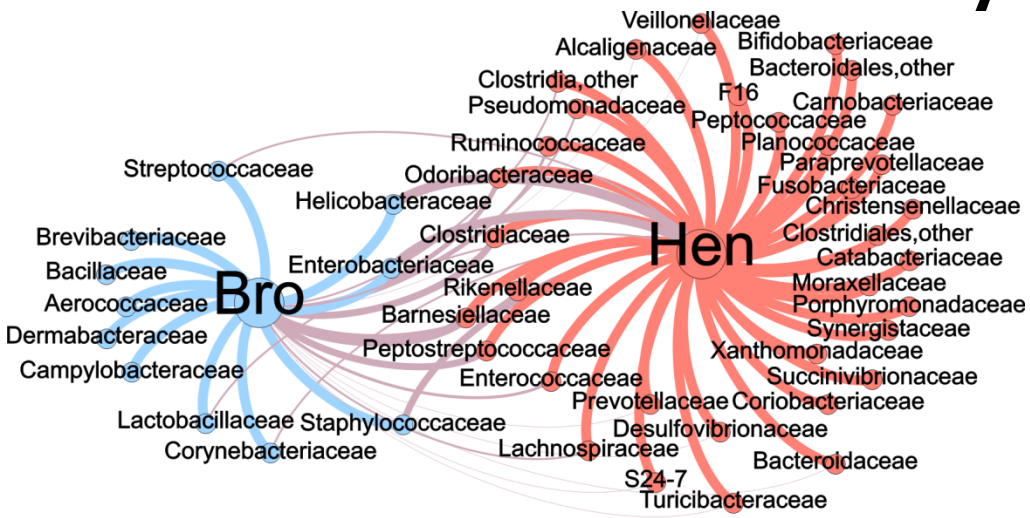

100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
K4 K7 K10 K13 K16 K19 I7 I10 I13 I16 I19 A-K7 A-K10 A-I10 C-K7 C-K10 C-I10
Lactobacillaceae
Ruminococcaceae
Lachnospiraceae
Enterobacteriaceae
Experiment 1
Experiment 2

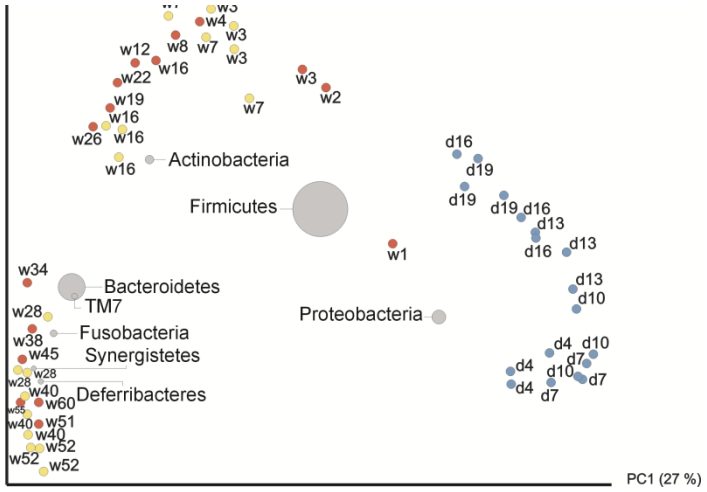

Ruminococcaceae
Rikenellaceae
Porphyromonadaceae
Prevotellaceae
Bacteroidaceae
Lachnospiraceae
1. fáze
2. fáze
3. fáze
4. fáze

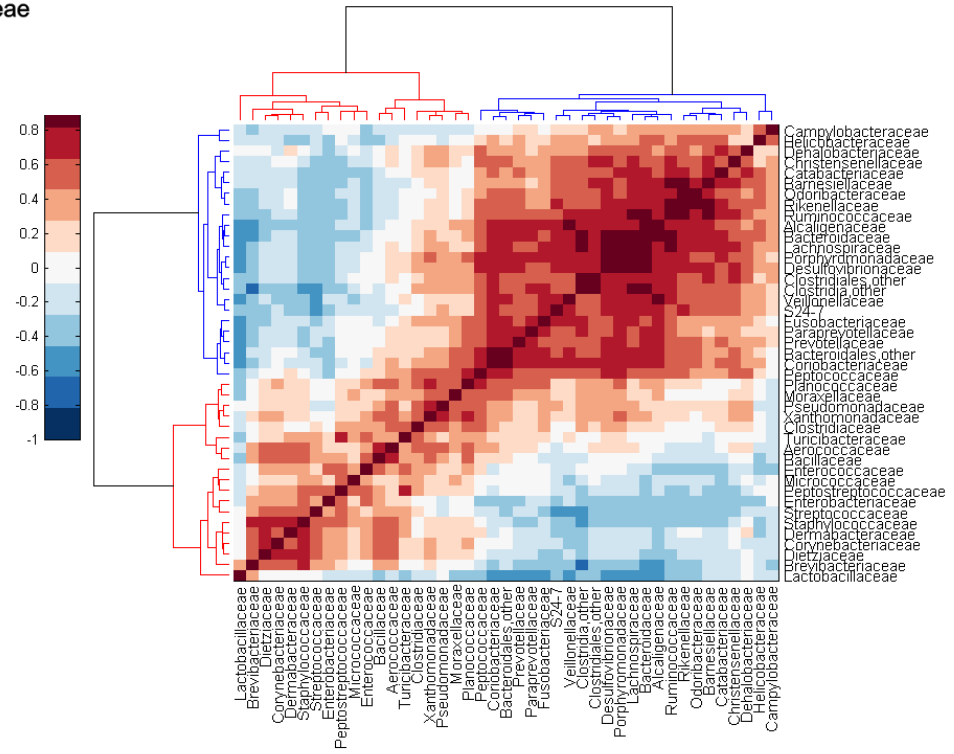

100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Enterobacteriaceae
Ruminococcaceae
Lachnospiraceae
Lactobacillaceae
Helicobacteraceae
Veillonelaceae
Rikenellaceae
Bacteroidaceae
kontrola
inokulace



# Analýza

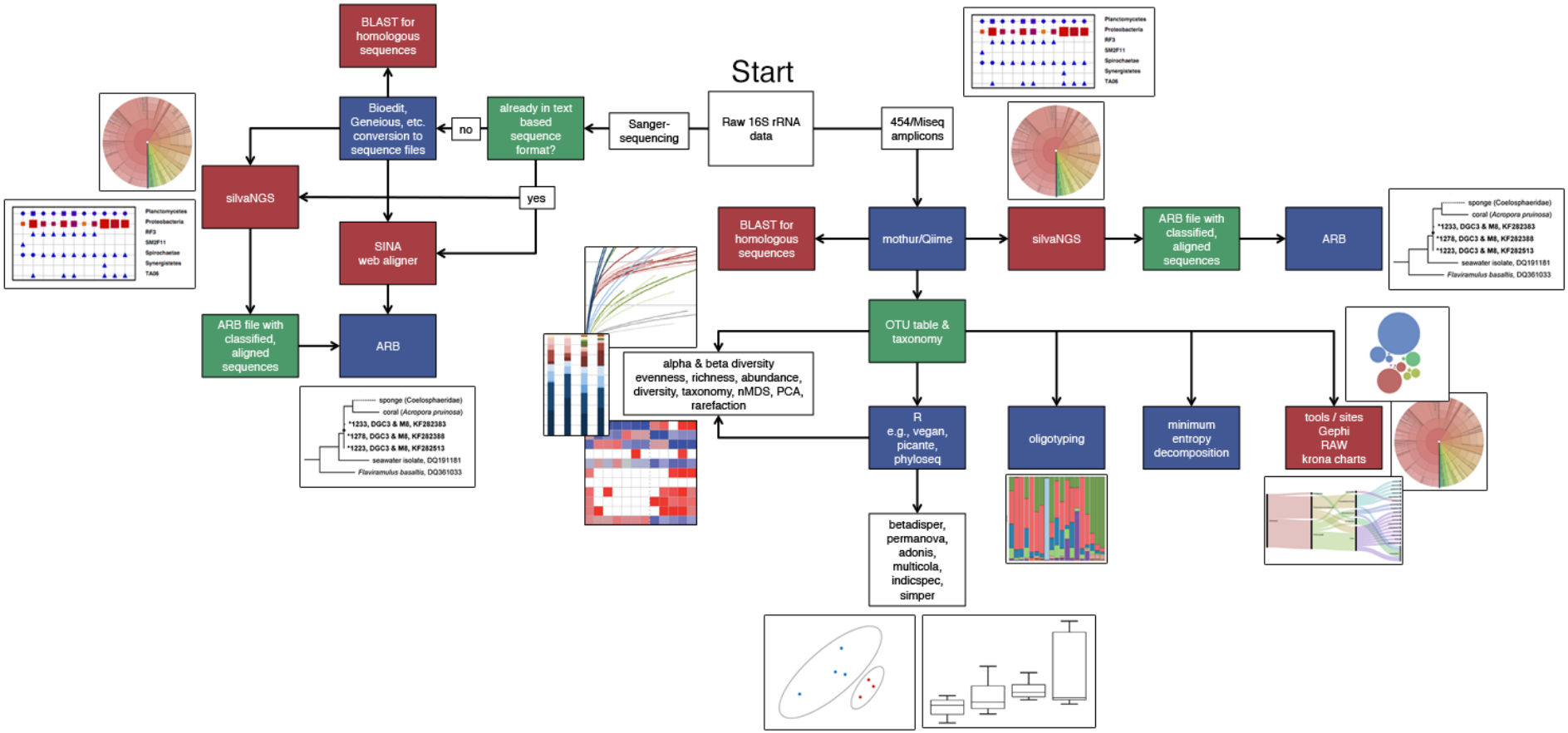


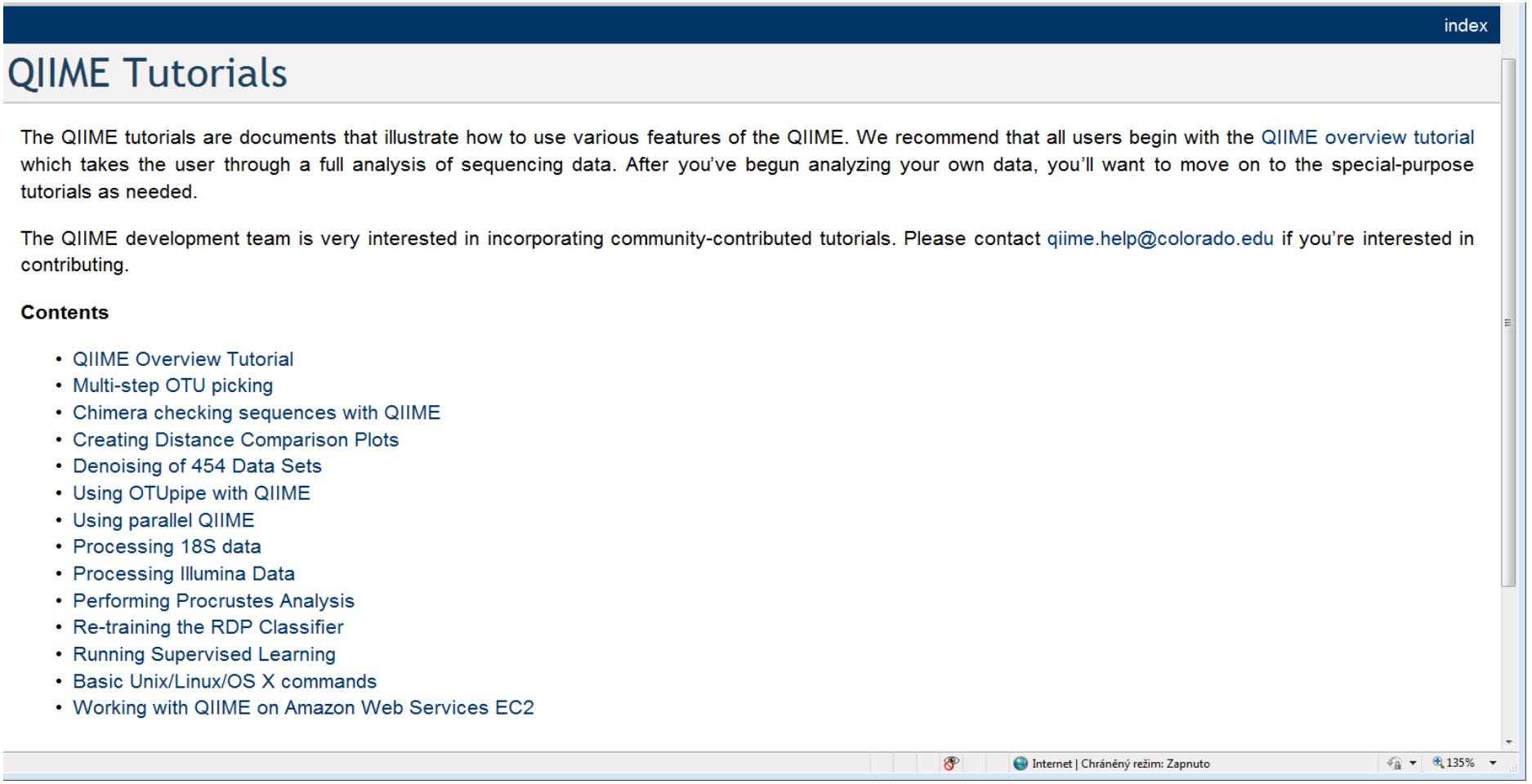

Research centre
for toxic compounds
in the environment

# Vyhodnocování – nejčastěji užívané pipelines

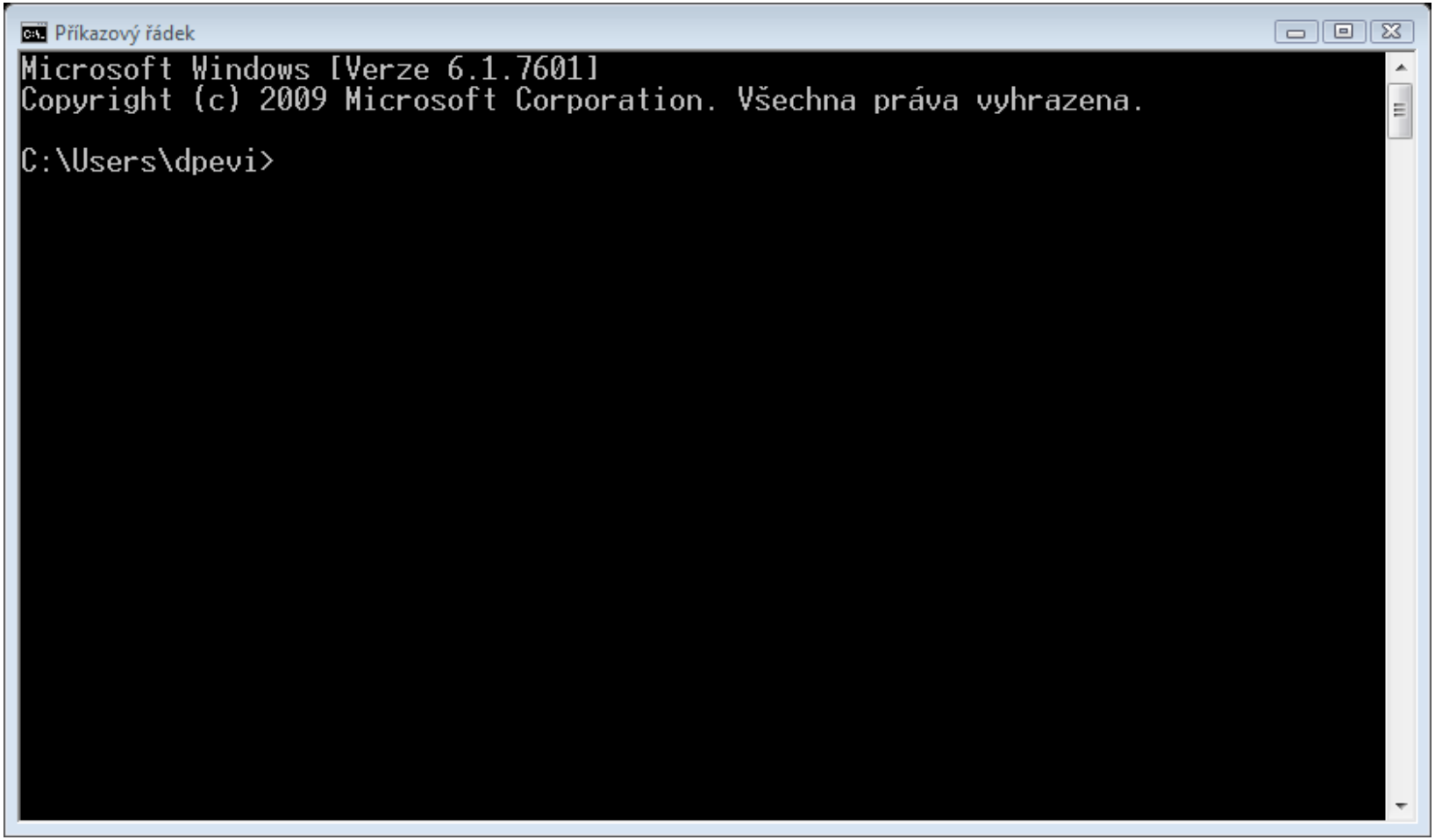


# Qiime

- http://www.qiime.org/svn_documentation/index.html
- Pracuje se v příkazovém řádku, nutné znát základní příkazy



# Příkazový řádek



# Příkazy

check_id_map.py -m mapa.txt -o mapping_output –v
quality_scores_plot.py -q seqs.qual -o quality_histograms/ -s 20

split_libraries.py -m mapa.txt -f IR -q IR -n 1000000 -o output_split_lib2/ -M 1 -b 13 –z truncate_only -l

sed '/^[A-Z]/s/[A-Za-z]\{50\}\([A-Za-z]\{340\}\).*/\1/' output_split_lib/seqs.fna > cutseq.fna

pick_otus.py -i cutseq.fna -o picked_otus/

pick_rep_set.py -i picked_otus/cutseq_otus.txt -f cutseq.fna

assign_taxonomy.py -i cutseq.fna_rep_set.fasta -c 0.5 -o assigned_taxonomy/

parallel_align_seqs_pynast.py -i cutseq.fna_rep_set.fasta -t core_set_aligned.fasta -O 6 -o parallel_align

parallel_identify_chimeric_seqs.py -m ChimeraSlayer -i parallel_align/cutseq.fna_rep_set_aligned.fasta -a core_set_aligned.fasta -o chimeric_seqs.txt -v

filter_fasta.py -f parallel_align/cutseq.fna_rep_set_aligned.fasta -o non_chimeric_rep_set_aligned.fasta -s chimeric_seqs.txt -n

make_otu_table.py -i otu_map.txt -o otu_table.biom -e chimeric_seqs.txt -t taxonomy.txt

# Ukázka příkazů

index

## *split_libraries.py* - Split libraries according to barcodes specified in mapping file

**Description:**

Since newer sequencing technologies provide many reads per run (e.g. the 454 GS FLX Titanium series can produce 400-600 million base pairs with 400-500 base pair read lengths) researchers are now finding it useful to combine multiple samples into a single 454 run. This multiplexing is achieved through the application of a pyrosequencing-tailored nucleotide barcode design (described in (Parameswaran et al., 2007)). By assigning individual, unique sample specific barcodes, multiple sequencing runs may be performed in parallel and the resulting reads can later be binned according to sample. The script split_libraries.py performs this task, in addition to several quality filtering steps including user defined cut-offs for: sequence lengths; end-trimming; minimum quality score. To summarize, by using the fasta, mapping, and quality files, the program split_libraries.py will parse sequences that meet user defined quality thresholds and then rename each read with the appropriate Sample ID, thus formatting the sequence data for downstream analysis. If a combination of different sequencing technologies are used in any particular study, split_libraries.py can be used to perform the quality-filtering for each library individually and the output may then be combined.

Sequences from samples that are not found in the mapping file (no corresponding barcode) and sequences without the correct primer sequence will be excluded. Additional scripts can be used to exclude sequences that match a given reference sequence (e.g. the human genome; exclude_seqs_by_blast.py) and/or sequences that are flagged as chimeras (identify_chimeric_seqs.py).

**Usage:** `split_libraries.py [options]`

**Input Arguments:**

```
[REQUIRED]

-m, --map
    Name of mapping file. NOTE: Must contain a header line indicating SampleID in the first column and BarcodeSequence in the second, LinkerPrimerSequence in the third.

-f, --fasta
    Names of fasta files, comma-delimited

[OPTIONAL]

-q, --qual
    Names of qual files, comma-delimited [default: None]

-r, --remove_unassigned
    DEPRECATED: pass -retain_unassigned_reads to keep unassigned reads [default: None]

-l, --min-seq-length
    Minimum sequence length, in nucleotides [default: 200]

-L, --max-seq-length
    Maximum sequence length, in nucleotides [default: 1000]

-t, --trim-seq-length
    Calculate sequence lengths after trimming primers and barcodes [default: False]

-s, --min-qual-score
    Min average qual score allowed in read [default: 25]

-k, --keep-primer
    Do not remove primer from sequences
```

# Ukázka příkazů

**Output:**

Three files are generated by split_libraries.py:

1. .fna file (e.g. seqs.fna) - This is a FASTA file containing all sequences which meet the user-defined parameters, where each sequence identifier now contains its corresponding sample id from mapping file.
2. histograms.txt- This contains the counts of sequences with a particular length.
3. split_library_log.txt - This file contains a summary of the split_libraries.py analysis. Specifically, this file includes information regarding the number of sequences that pass quality control (number of seqs written) and how these are distributed across the different samples which, through the use of bar-coding technology, would have been pooled into a single 454 run. The number of sequences that pass quality control will depend on length restrictions, number of ambiguous bases, max homopolymer runs, barcode check, etc. All of these parameters are summarized in this file. If raw sequences do not meet the specified quality thresholds they will be omitted from downstream analysis. Since we never see a perfect 454 sequencing run, the number of sequences written should always be less than the number of raw sequences. The number of sequences that are retained for analysis will depend on the quality of the 454 run itself in addition to the default data filtering thresholds in the split_libraries.py script. The default parameters (minimum quality score = 25, minimum/maximum length = 200/1000, no ambiguous bases allowed, no mismatches allowed in primer sequence) can be adjusted to meet the user's needs.

**Standard Example:**

Using a single 454 run, which contains a single FASTA, QUAL, and mapping file while using default parameters and outputting the data into the Directory "Split_Library_Output":

```
split_libraries.py -m Mapping_File.txt -f 1.TCA.454Reads.fna -q 1.TCA.454Reads.qual -o Split_Library_Output/
```

**Multiple FASTA and QUAL Files Example:**

For the case where there are multiple FASTA and QUAL files, the user can run the following comma-separated command as long as there are not duplicate barcodes listed in the mapping file:

```
split_libraries.py -m Mapping_File.txt -f 1.TCA.454Reads.fna,2.TCA.454Reads.fna -q 1.TCA.454Reads.qual,2.TCA.454Reads.qual -o Split_Library_Output_comma_separated/
```

**Duplicate Barcode Example:**

An example of this situation would be a study with 1200 samples. You wish to have 400 samples per run, so you split the analysis into three runs and reuse barcoded primers (you only have 600). After initial analysis you determine a small subset is underrepresented (<500 sequences per samples) and you boost the number of sequences per sample for this subset by running a fourth run. Since the same sample IDs are in more than one run, it is likely that some sequences will be assigned the same unique identifier by split_libraries.py when it is run separately on the four different runs, each with their own barcode file. This will cause a problem in file concatenation of the four different runs into a single large file. To avoid this, you can use the '-n' parameter which defines a start index for split_libraries.py. From experience, most FLX runs (when combining both files for a single plate) will have 350,000 to 650,000 sequences. Thus, if Run 1 for split_libraries.py uses '-n 1000000', Run 2 uses '-n 2000000', etc., then you are guaranteed to have unique identifiers after concatenating the results of multiple FLX runs. With newer technologies you will just need to make sure that your start index spacing is greater than the potential number of sequences.

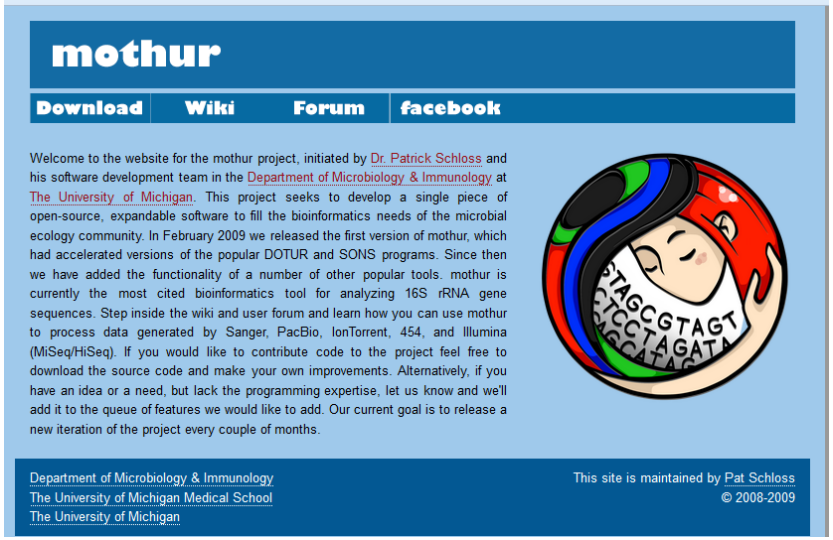
To run split_libraries.py, you will need two or more (depending on the number of times the barcodes were reused) separate mapping files (one for each Run, for example one for Run1 and another one for Run2), then you can run split_libraries.py using the FASTA and mapping file for Run1 and FASTA and mapping file for Run2. Once you have run split libraries on each file independently, you can concatenate (e.g. using the 'cat' command) the sequence files that were generated by split_libraries.py. You can also concatenate the mapping files, since the barcodes are not necessary for downstream analyses, unless the same sample IDs are found in multiple mapping files.

Run split_libraries.py on Run 1:

```
split_libraries.py -m Mapping_File.txt -f 1.TCA.454Reads.fna -q 1.TCA.454Reads.qual -o Split_Library_Run1_Output/ -n 1000000
```

# mothur

- http://www.mothur.org/




https://www.youtube.com/watch?v=X4aV4J8FkEU&nohtml5=False

# Qiime vs. mothur

- http://blog.mothur.org/2016/01/12/mothur-and-qiime/

## mothur and QIIME

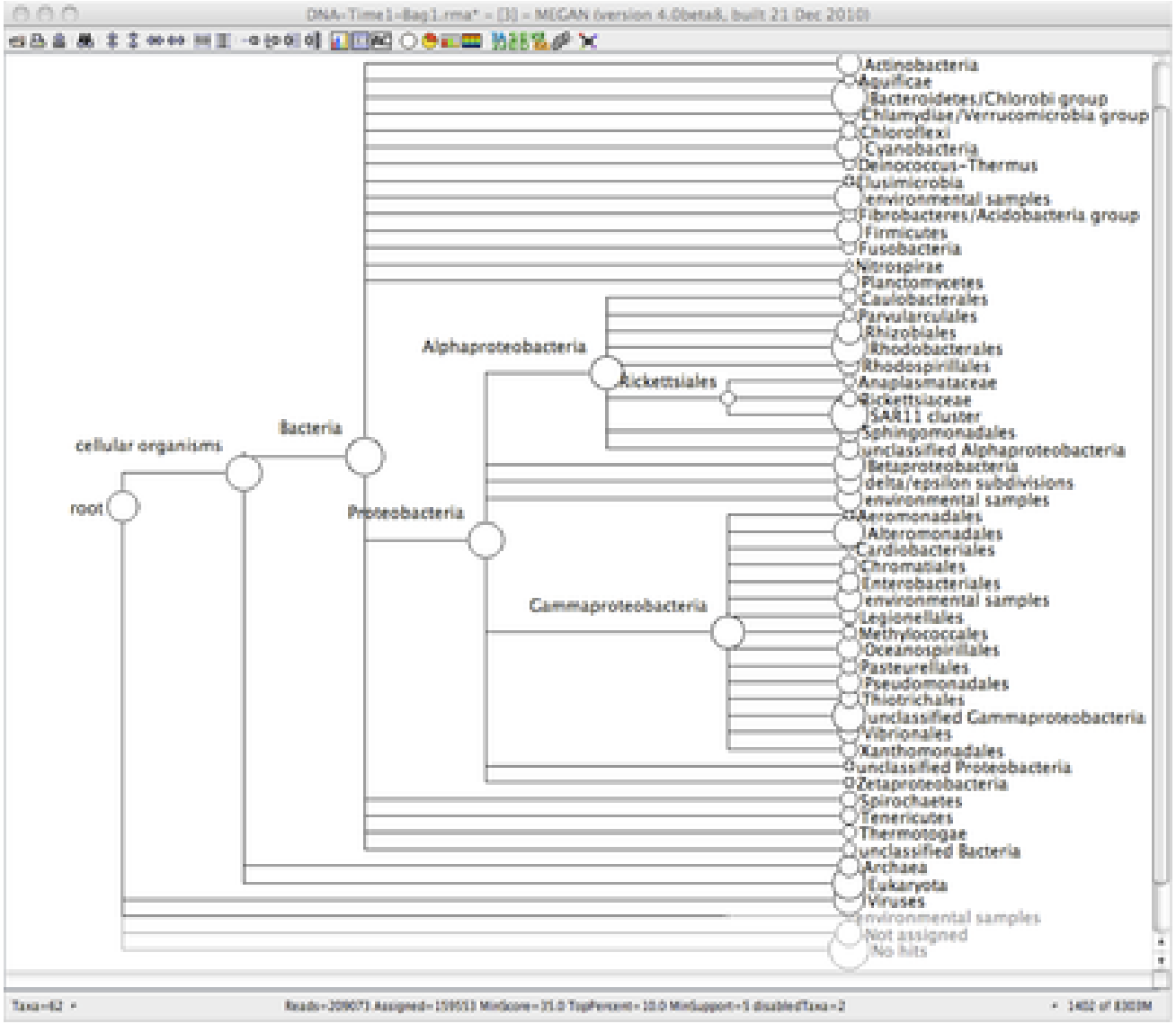
Jan 12, 2016 • PD Schloss • 32 min read

Despite their differences in philosophy, *most* of the differences in mothur and QIIME are cosmetic. Both packages have been successful. Having both of them around is good for microbial ecology. Within both packages there are warts - inconveniences to the users and antiquated/bad ideas. Within both packages there are strengths. If you are going to criticize someone for their choice of software, do it for some specific point. If you are going to campaign for mothur or QIIME, do your best to accurately represent the strengths of your pet package.

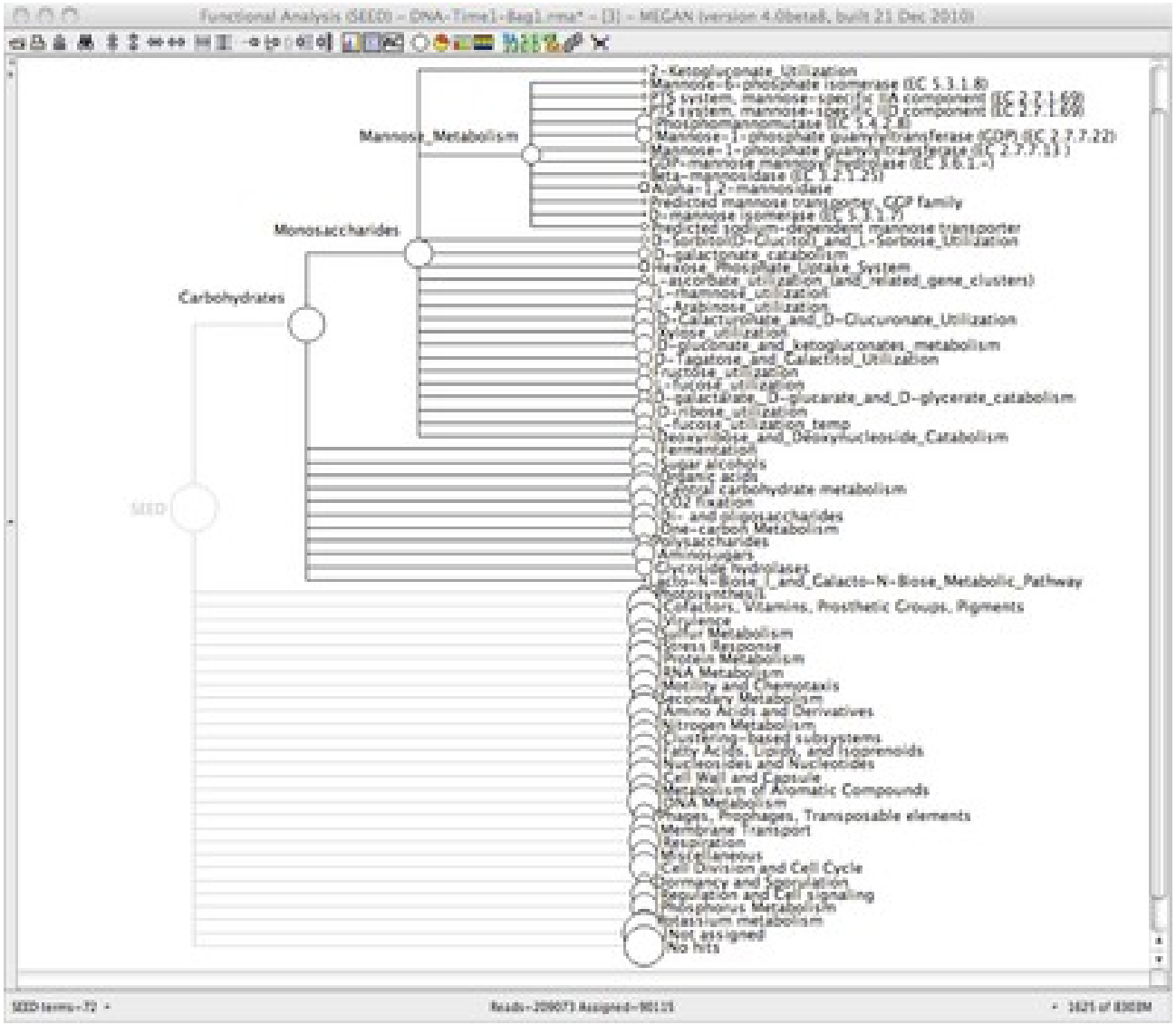
# Megan

- http://ab.inf.uni-tuebingen.de/software/megan5/

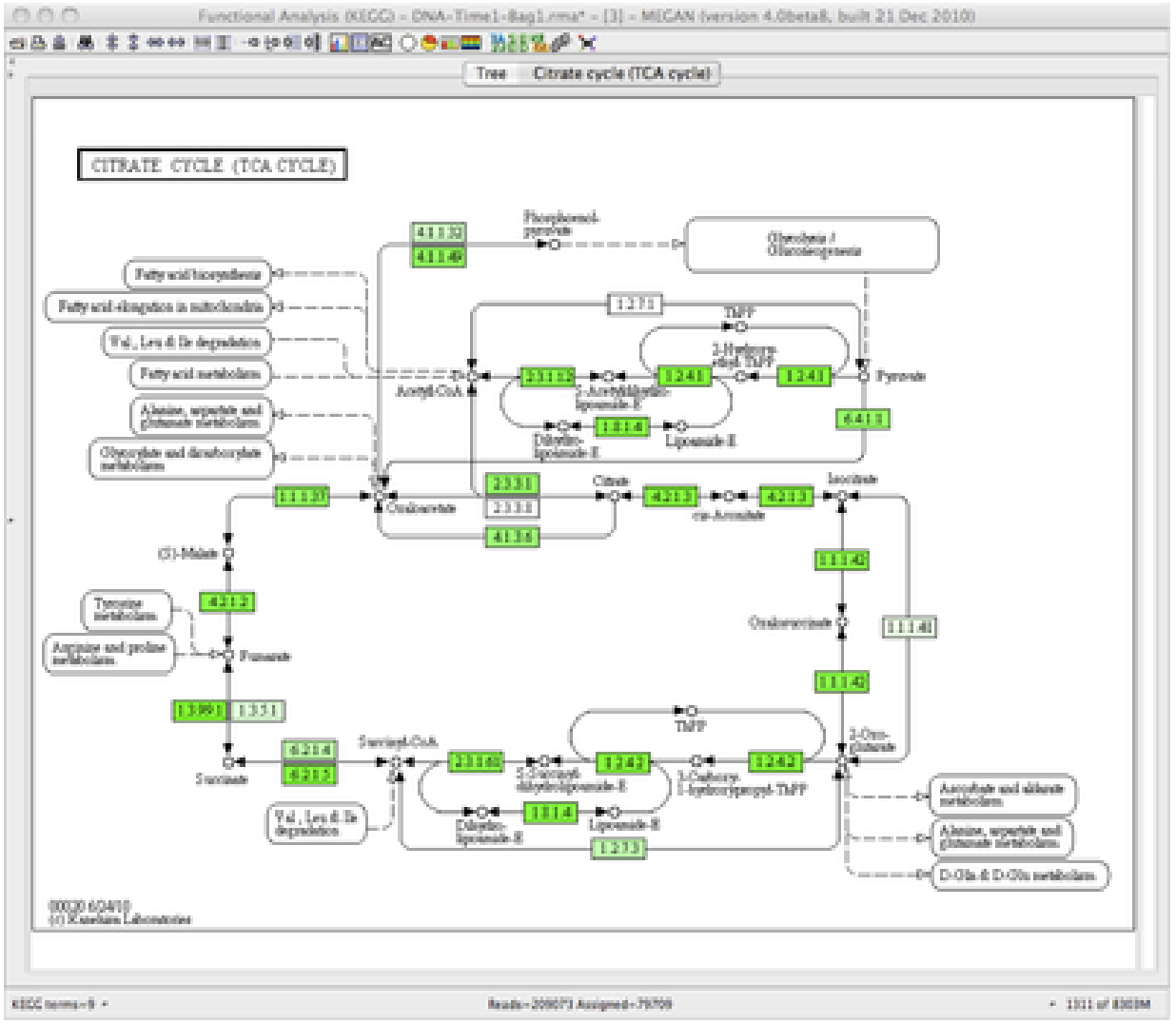
# Taxonomická analýza



# SEED analýza

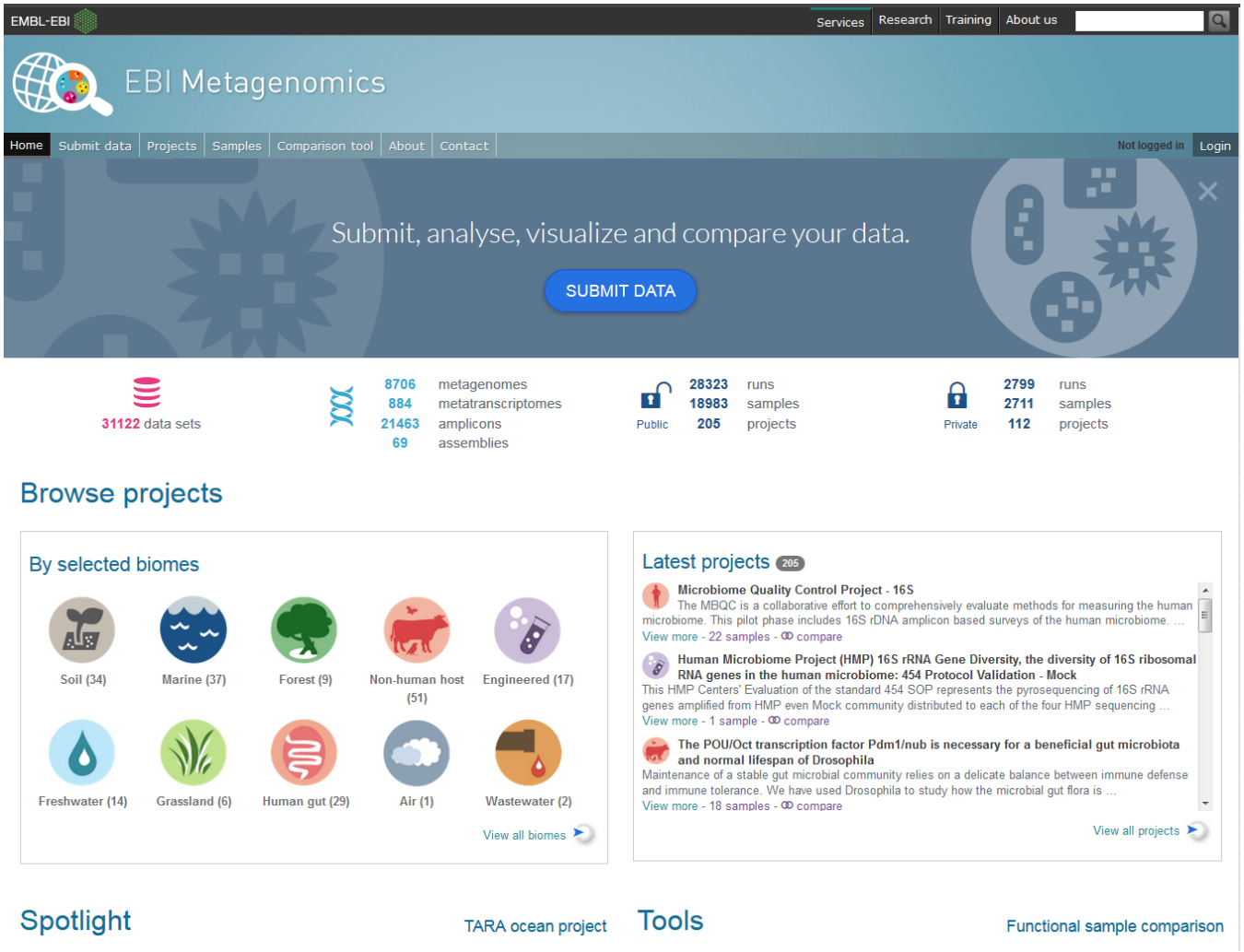


# KEGG analýza



# EBI Metagenomic

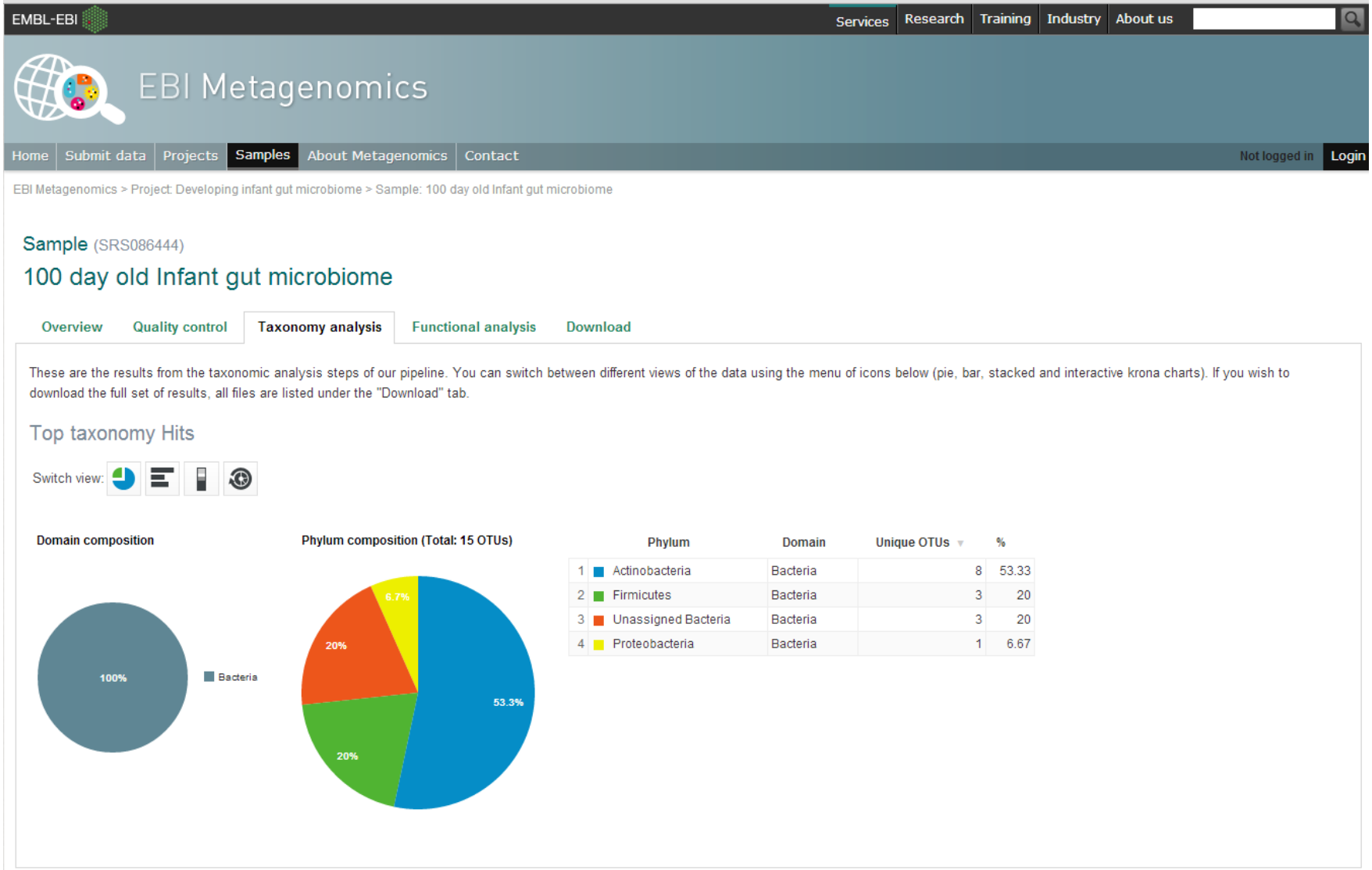
https://www.ebi.ac.uk/metagenomics/




https://www.ebi.ac.uk/metagenomics/projects/SRP000319/samples/SRS000998/runs/SRR029687/results/versions/1.0

# EBI pipeline

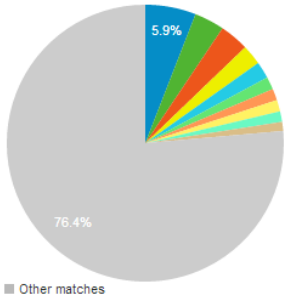
# Taxonomická analýza



# Funkční analýza

## InterPro match summary

**InterPro matches summary (Total: 781)**




Filter table:

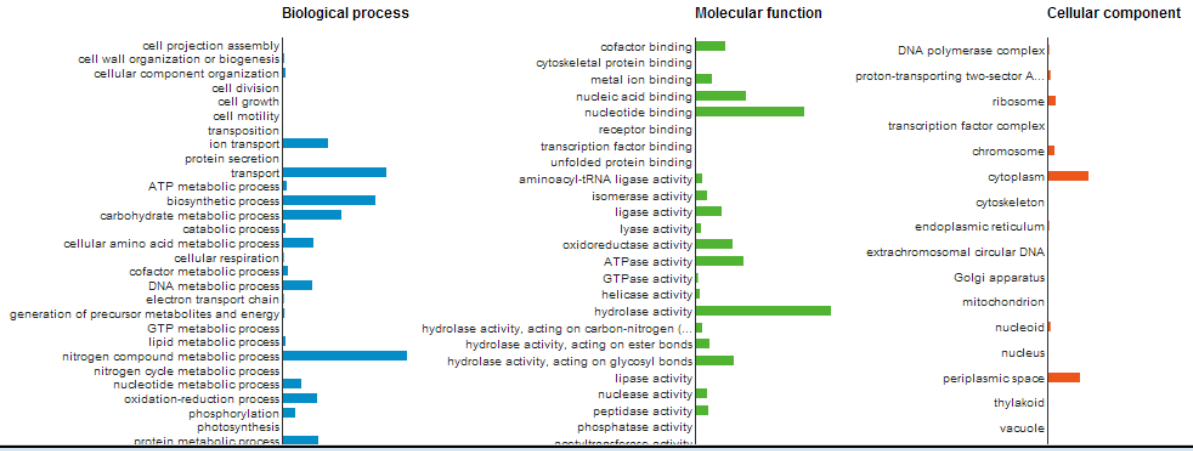
| | Entry name | ID | pCDS matched | % |
|---|---|---|---|---|
| 1 | NAD(P)-binding domain | IPR016040 | 184 | 5.91 |
| 2 | Regulator of K+ conductance, N-terminal | IPR003148 | 109 | 3.5 |
| 3 | ABC transporter-like | IPR003439 | 106 | 3.41 |
| 4 | Potassium uptake protein TrkA | IPR006036 | 73 | 2.35 |
| 5 | Glycoside hydrolase, family 3, N-terminal | IPR001764 | 61 | 1.96 |
| 6 | Extracellular solute-binding protein, family 3 | IPR001638 | 46 | 1.48 |
| 7 | Winged helix-turn-helix transcription repressor DNA-binding | IPR011991 | 42 | 1.35 |
| 8 | Aldolase-type TIM barrel | IPR013785 | 42 | 1.35 |
| 9 | Lacto-N-biose phosphorylase | IPR012711 | 38 | 1.22 |
| 10 | Serpin domain | IPR023796 | 33 | 1.06 |

prev next 1 2 10 70 79

## GO Terms annotation

A summary of Gene Ontology (GO) terms derived from InterPro matches to your sample is provided in the charts below.

Switch view:

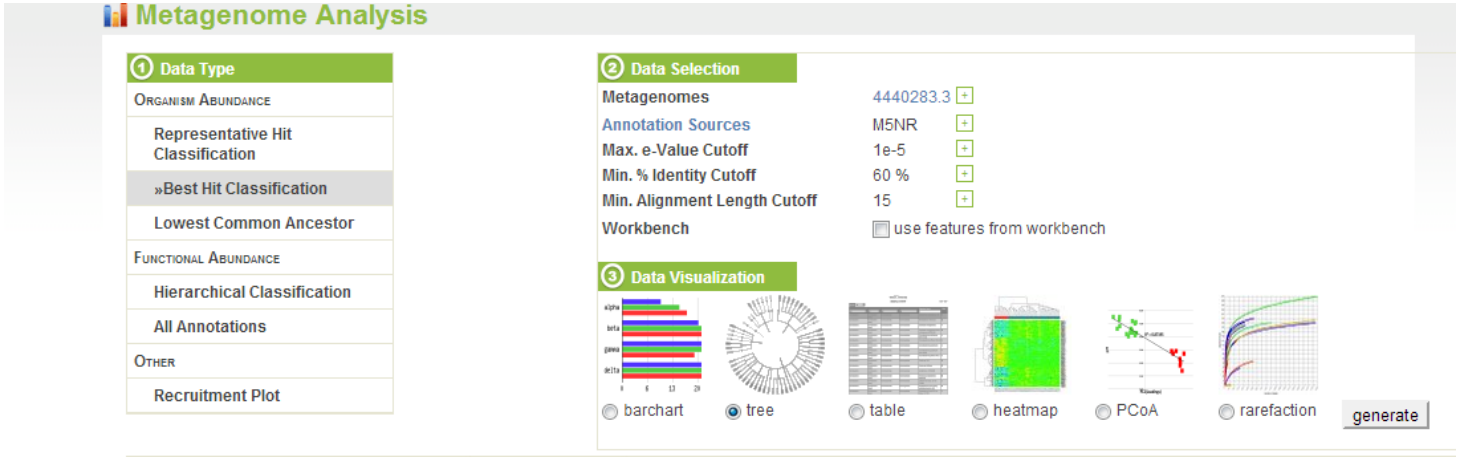


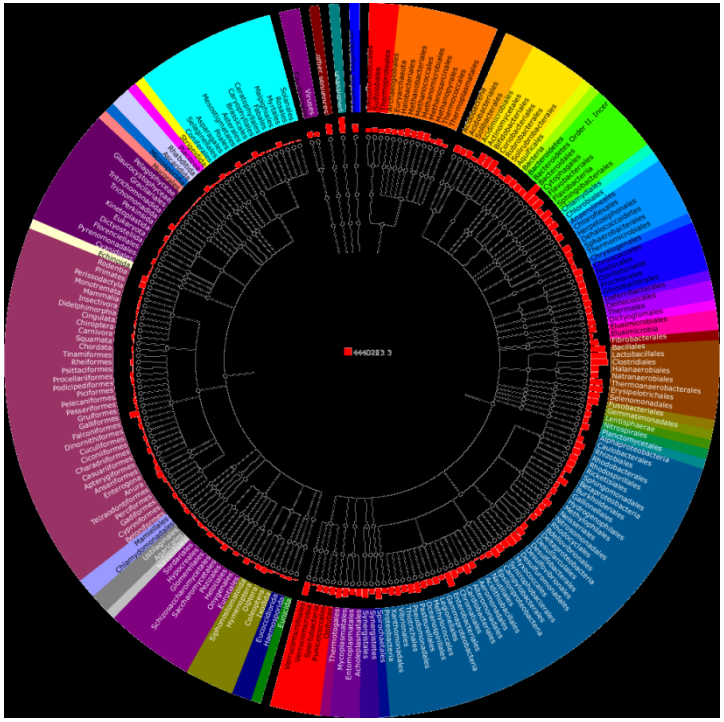
# MG - RAST

http://metagenomics.anl.gov/

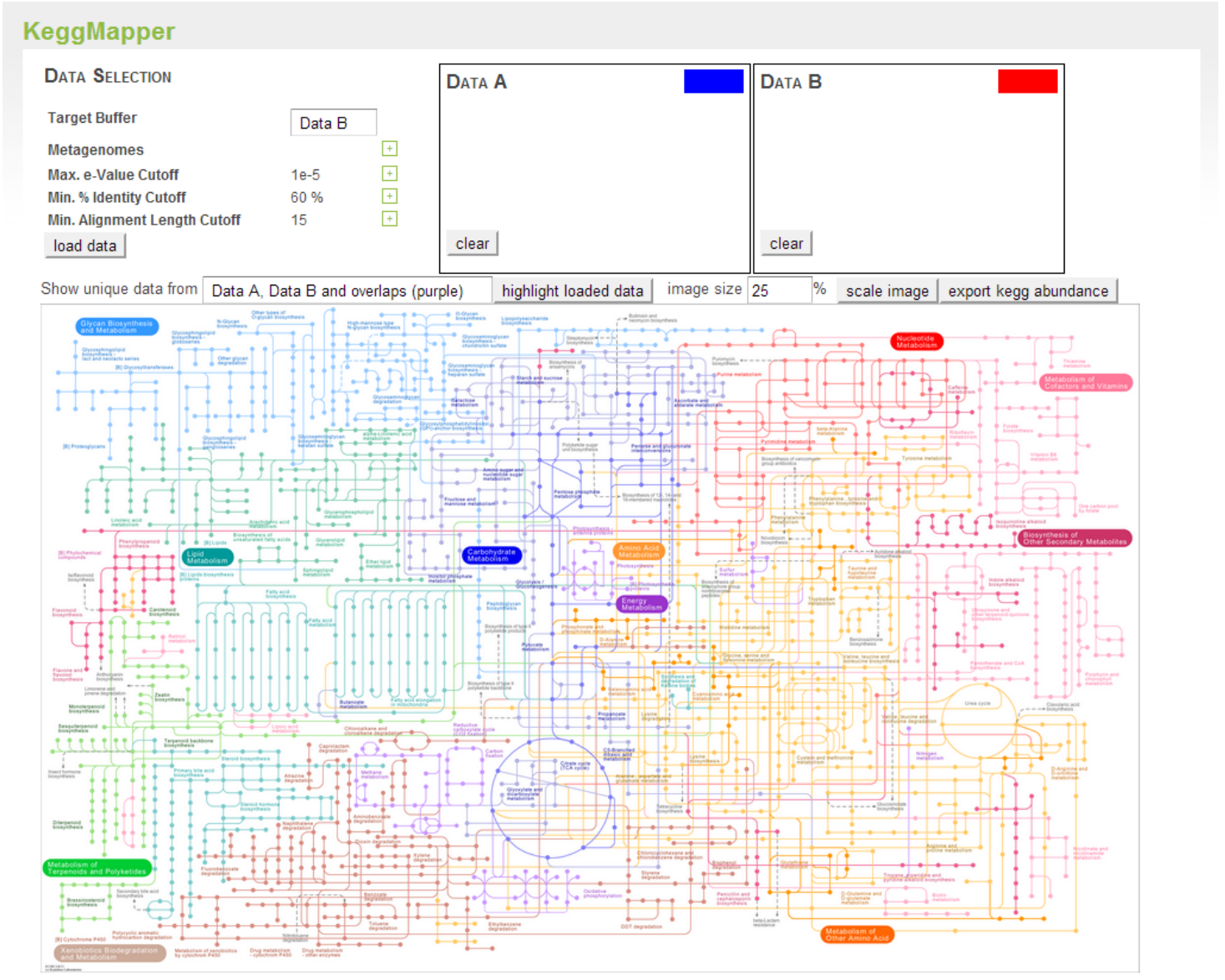

http://metagenomics.anl.gov/metagenomics.cgi?page=MetagenomeOverview&metagenome=4447943.3

# Taxonomická analýza



# KEGG analýza



# Databáze genu pro 16S rRNA

# RDP

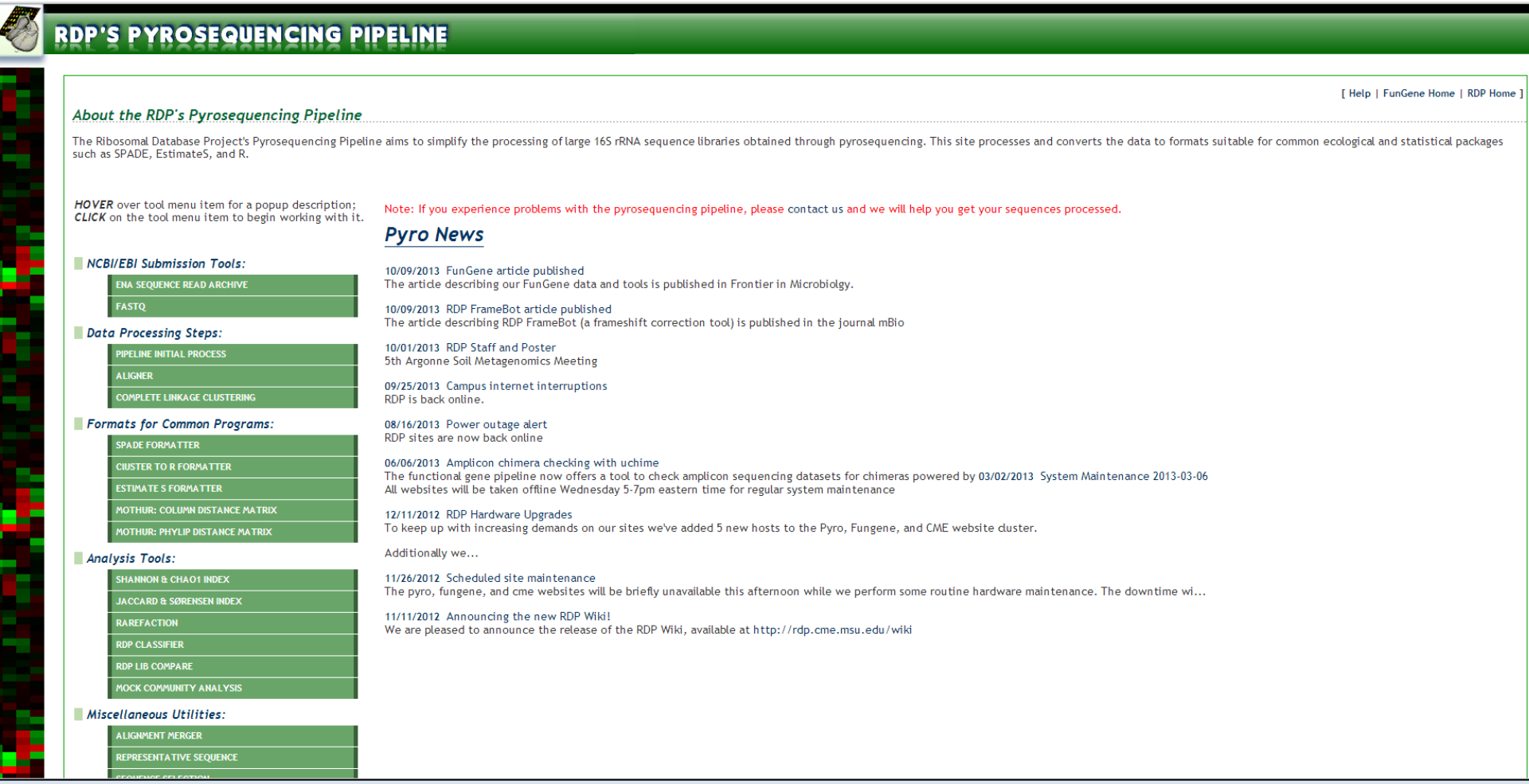
## http://rdp.cme.msu.edu


http://rdp.cme.msu.edu/tutorials/init_process/RDPtutorial_INITIAL-PROCESS.html

http://rdp.cme.msu.edu/tutorials/init_process/RDPtutorial_INITIAL-PROCESS_pe.html

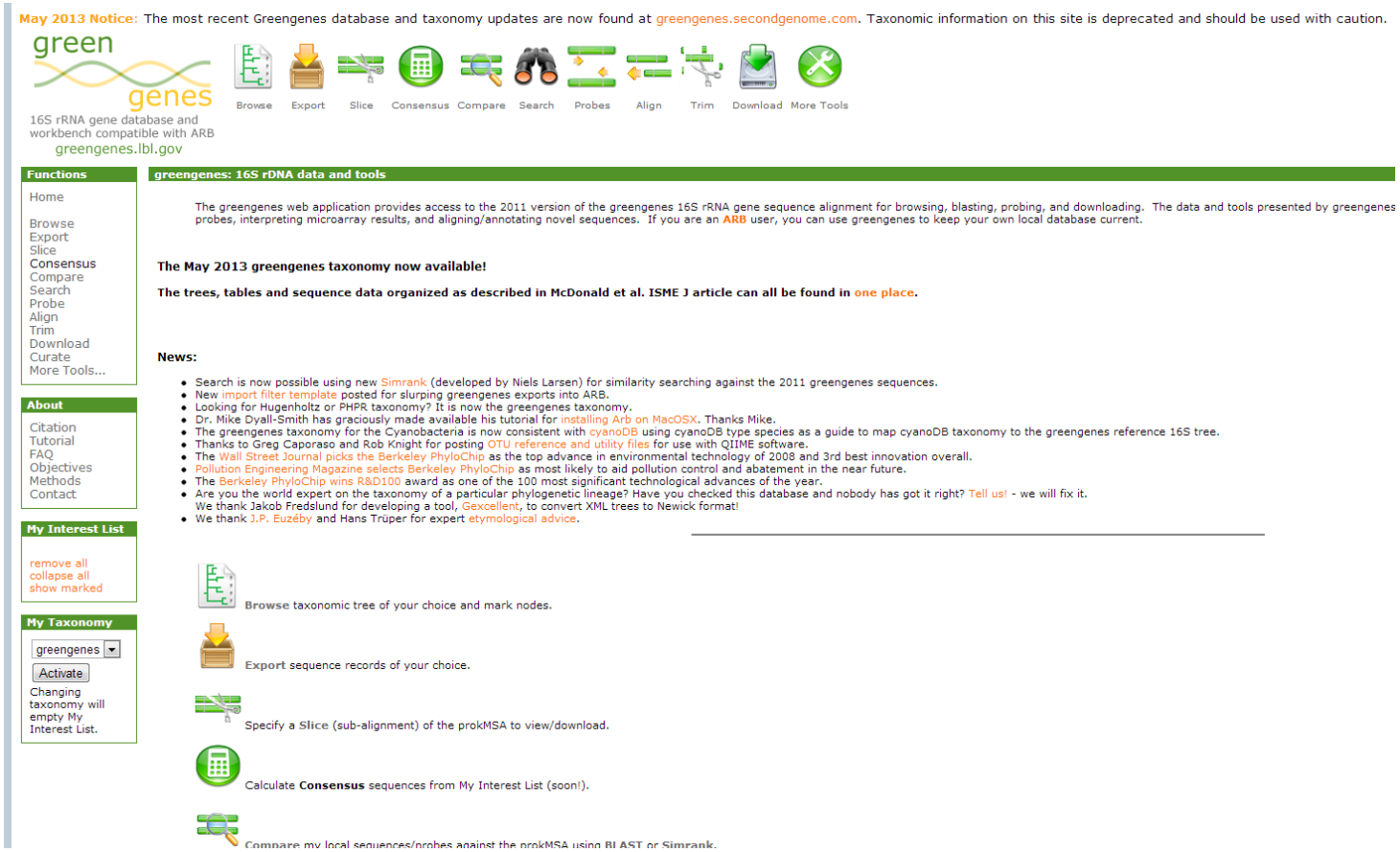
http://rdp.cme.msu.edu/tutorials/Submission_Tools/fastq.html

# RDP


RDP'S PYROSEQUENCING PIPELINE
[ Help | FunGene Home | RDP Home ]
About the RDP's Pyrosequencing Pipeline
The Ribosomal Database Project's Pyrosequencing Pipeline aims to simplify the processing of large 16S rRNA sequence libraries obtained through pyrosequencing. This site processes and converts the data to formats suitable for common ecological and statistical packages such as SPADE, EstimateS, and R.
HOVER over tool menu item for a popup description;
CLICK on the tool menu item to begin working with it.
Note: If you experience problems with the pyrosequencing pipeline, please contact us and we will help you get your sequences processed.
Pyro News
NCBI/EBI Submission Tools:
ENA SEQUENCE READ ARCHIVE
FASTQ
Data Processing Steps:
PIPELINE INITIAL PROCESS
ALIGNER
COMPLETE LINKAGE CLUSTERING
Formats for Common Programs:
SPADE FORMATTER
CLUSTER TO R FORMATTER
ESTIMATE S FORMATTER
MOTHUR: COLUMN DISTANCE MATRIX
MOTHUR: PHYLIP DISTANCE MATRIX
Analysis Tools:
SHANNON & CHAO1 INDEX
JACCARD & SØRENSEN INDEX
RAREFACTION
RDP CLASSIFIER
RDP LIB COMPARE
MOCK COMMUNITY ANALYSIS
Miscellaneous Utilities:
ALIGNMENT MERGER
REPRESENTATIVE SEQUENCE
10/09/2013 FunGene article published
The article describing our FunGene data and tools is published in Frontier in Microbiolgy.
10/09/2013 RDP FrameBot article published
The article describing RDP FrameBot (a frameshift correction tool) is published in the journal mBio
10/01/2013 RDP Staff and Poster
5th Argonne Soil Metagenomics Meeting
09/25/2013 Campus internet interruptions
RDP is back online.
08/16/2013 Power outage alert
RDP sites are now back online
06/06/2013 Amplicon chimera checking with uchime
The functional gene pipeline now offers a tool to check amplicon sequencing datasets for chimeras powered by 03/02/2013 System Maintenance 2013-03-06
All websites will be taken offline Wednesday 5-7pm eastern time for regular system maintenance
12/11/2012 RDP Hardware Upgrades
To keep up with increasing demands on our sites we've added 5 new hosts to the Pyro, Fungene, and CME website cluster.
Additionally we...
11/26/2012 Scheduled site maintenance
The pyro, fungene, and cme websites will be briefly unavailable this afternoon while we perform some routine hardware maintenance. The downtime wi...
11/11/2012 Announcing the new RDP Wiki!
We are pleased to announce the release of the RDP Wiki, available at http://rdp.cme.msu.edu/wiki

# GreenGenes

## http://greengenes.lbl.gov




http://greengenes.lbl.gov/cgi-bin/JD_Tutorial/nph-Tutorial_2Main2.cgi

# Silva

- http://www.arb-silva.de/

# Srovnání databází

The ISME Journal (2011), 1–10

www.nature.com/ismej

**ORIGINAL ARTICLE**

## Impact of training sets on classification of high-throughput bacterial 16s rRNA gene surveys

Jeffrey J Werner[1,7], Omry Koren[2,7], Philip Hugenholtz[3], Todd Z DeSantis[4], William A Walters[5], J Gregory Caporaso[5], Largus T Angenent[1], Rob Knight[5,6] and Ruth E Ley[2]

[1]*Department of Biological and Environmental Engineering, Cornell University, Ithaca, NY, USA;* [2]*Department of Microbiology, Cornell University, Ithaca, NY, USA;* [3]*Australian Centre for Ecogenomics, School of Chemistry and Molecular Biosciences, The University of Queensland, Brisbane, QLD, Australia;* [4]*Center for Environmental Biotechnology, Lawrence Berkeley National Laboratory, Berkeley, CA, USA;* [5]*Department of Biochemistry and Chemistry, University of Colorado, Boulder, CO, USA and* [6]*Howard Hughes Medical Institute, University of Colorado, Boulder, CO, USA*

**Taxonomic classification of the thousands–millions of 16S rRNA gene sequences generated in microbiome studies is often achieved using a naïve Bayesian classifier (for example, the Ribosomal Database Project II (RDP) classifier), due to favorable trade-offs among automation, speed and accuracy. The resulting classification depends on the reference sequences and taxonomic hierarchy used to train the model; although the influence of primer sets and classification algorithms have been explored in detail, the influence of training set has not been characterized. We compared classification results obtained using three different publicly available databases as training sets, applied to five different bacterial 16S rRNA gene pyrosequencing data sets generated (from human body, mouse gut, python gut, soil and anaerobic digester samples). We observed numerous advantages to using the largest, most diverse training set available, that we constructed from the Greengenes (GG) bacterial/archaeal 16S rRNA gene sequence database and the latest GG taxonomy. Phylogenetic clusters of previously unclassified experimental sequences were identified with notable improvements (for example, 50% reduction in reads unclassified at the phylum level in mouse gut, soil and anaerobic digester samples), especially for phylotypes belonging to specific phyla (Tenericutes, Chloroflexi, Synergistetes and Candidate phyla TM6, TM7). Trimming the reference sequences to the primer region resulted in systematic improvements in classification depth, and greatest gains at higher confidence thresholds. Phylotypes unclassified at the genus level represented a greater proportion of the total community variation than classified operational taxonomic units in mouse gut and anaerobic digester samples, underscoring the need for greater diversity in existing reference databases.**

*The ISME Journal* advance online publication, 30 June 2011; doi:10.1038/ismej.2011.82
**Subject Category:** integrated genomics and post-genomics approaches in microbial ecology
**Keywords:** Greengenes; microbiome; naïve Bayesian classifier; pyrosequencing; taxonomy

# Srovnání databází



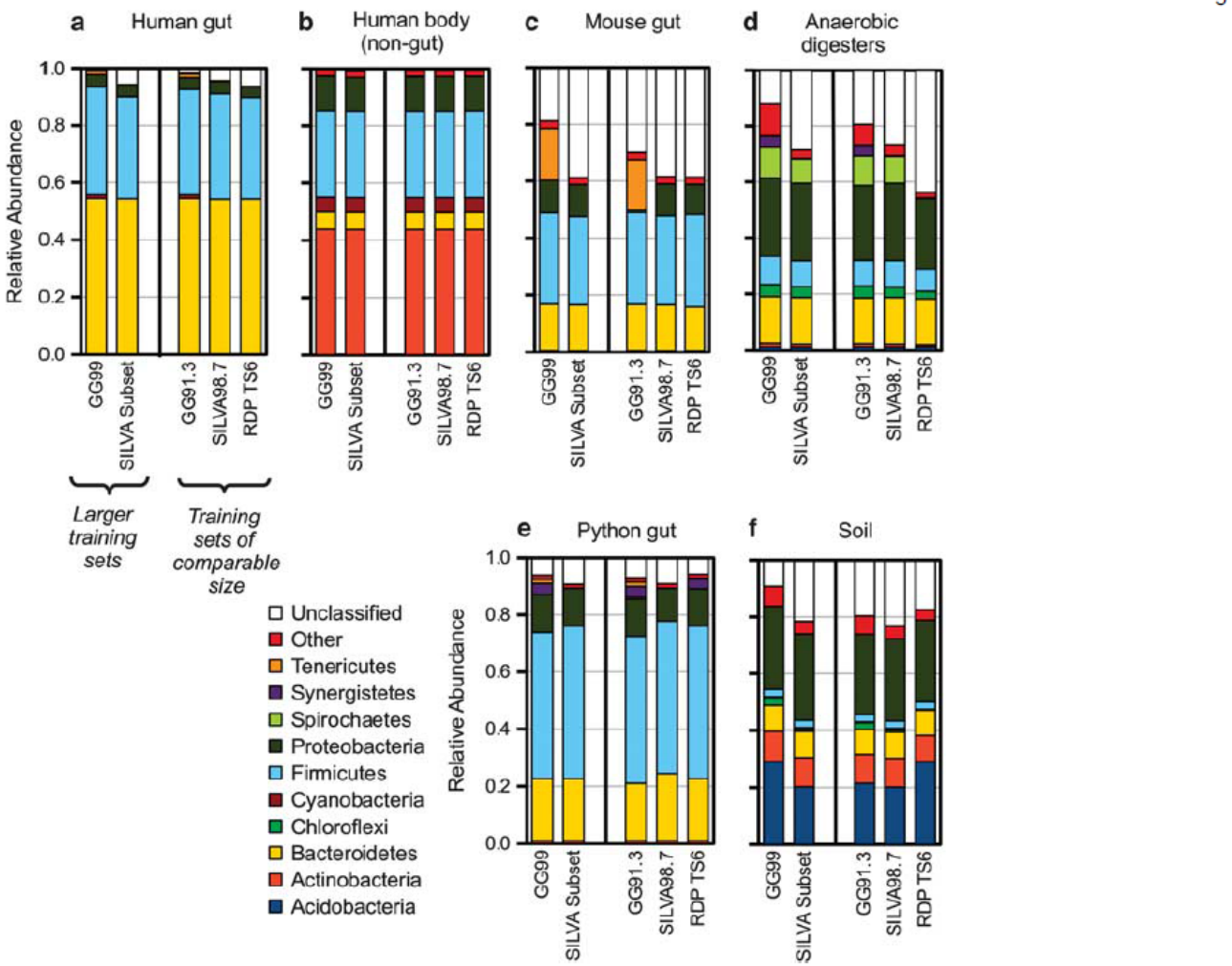



**Figure 1** Relative abundance of the 10 major phyla identified by naïve Bayesian classification using five different training sets: three of approximately the same size: GG91.3, SILVA98.7, and RDP TS6, and two larger training sets: GG99 and the SILVA subset for Mothur. Relative abundances were averaged for samples of five different studies (note that human gut is shown apart from non-gut samples from the sample study): (**a**) human gut, (**b**) non-gut human body locations, (**c**) mouse gut, (**d**) anaerobic digester, (**e**) python gut and (**f**) soils.

# Srovnání databází



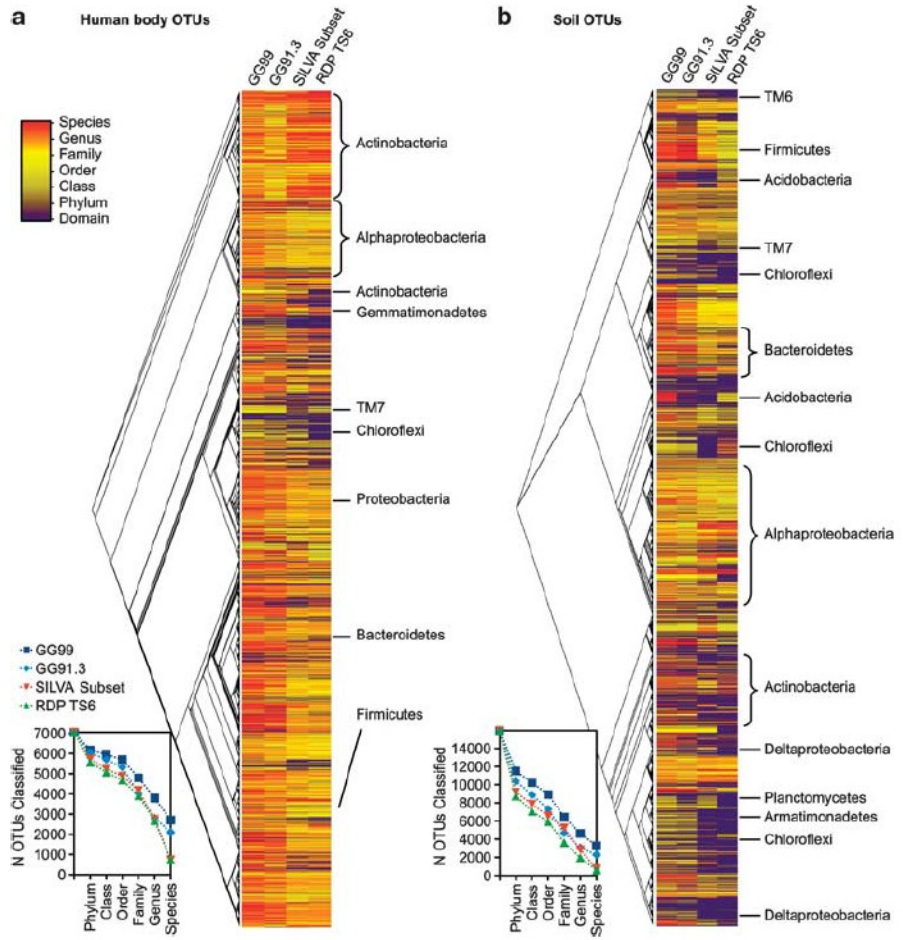



**Figure 2** Summary of OTU classification depth using each of the three training sets for two of the four studies: (**a**) human body OTUs, and (**b**) soil OTUs (other three data sets shown in Supplementary Figure S6). OTUs are organized according to evolutionary history, as determined by the FastTree approximately-maximum-likelihood tree constructed in the default QIIME pipeline. Inset charts summarize the total number of OTUs classified at each taxonomic level (GG99 = dark blue, GG91.3 = light blue, SILVA = green, RDP TS6 = orange).

# Další vyhodnocovací programy

REVIEW

## Metagenomic Surveys of Gut Microbiota

Rahul Shubhra Mandal [1,a], Sudipto Saha [2,*,b], Santasabuj Das [1,3,*,c]

[1] Biomedical Informatics Centre, National Institute of Cholera and Enteric Diseases, Kolkata 700010, India
[2] Bioinformatics Centre, Bose Institute, Kolkata 700054, India
[3] Division of Clinical Medicine, National Institute of Cholera and Enteric Diseases, Kolkata 700010, India

Received 8 July 2014; revised 10 February 2015; accepted 26 February 2015
Available online 13 July 2015

Handled by Fangqing Zhao

**Table 3 Tools/webservers related to gut microbiota studies**

| Name | Platform | Website | Main features | Ref. |
|---|---|---|---|---|
| QIIME | Stand alone | http://qiime.sourceforge.net/ | Network analysis, histograms of within- or between-sample diversity | [82] |
| mothur | Stand alone | http://www.mothur.org/ | Fast processing of large sequence data | [83] |
| RAMMCAP | Stand alone | http://weizhonglab.ucsd.edu/rammcap/cgibin/rammcap.cgi | Ultra fast sequence clustering and protein family annotation | [84] |
| MEGAN | Stand alone | http://www-ab.informatik.uni-tuebingen.de/software/megan/ | Laptop analysis of large metagenomic shotgun sequencing data sets | [85] |
| MetaPhlAn | Stand alone | http://huttenhower.sph.harvard.edu/metaphlan | Faster profiling of the composition of microbial communities using unique clade-specific marker genes | [86] |
| MetaVelvet | Stand alone | http://metavelvet.dna.bio.keio.ac.jp/ | High quality metagenomic assembler | [87] |
| SOAPdenovo2 | Stand alone | http://soap.genomics.org.cn/soapdenovo.html | Metagenomic assembler, specifically for Illumina GA short reads | [88] |
| MOCAT | Stand alone | http://vmlux.embl.de/~kultima/MOCAT/ | Generate taxonomic profiles and assemble metagenomes | [89] |
| SmashCommunity | Stand alone | http://www.bork.embl.de/software/smash/ | Performs assembly and gene prediction mainly for data from Sanger and 454 sequencing technologies | [90] |
| HUMAnN | Stand alone | http://huttenhower.sph.harvard.edu/humann | Analysis of metagenomic shotgun data from the Human Microbiome Project | [91] |
| FANTOM | Stand alone | http://www.sysbio.se/Fantom/ | Comparative analysis of metagenomics abundance data integrated with databases like KEGG Orthology, COG, PFAM and TIGRFAM, *etc.* | [92] |
| MetaCV | Stand alone | http://metacv.sourceforge.net/ | Classification short metagenomic reads (75–100 bp) into specific taxonomic | [94] |
| Phymm | Stand alone | http://www.cbcb.umd.edu/software/phymm/ | Phylogenetic classification of metagenomic short reads using interpolated Markov models | [97] |
| PhyloPythiaS | Web server | http://binning.bioinf.mpiinf.mpg.de/ | Fast and accurate sequence composition-based classifier that utilizes the hierarchical relationships between clades | [96] |
| TETRA | Web server | http://www.megx.net/tetra | Correlation of tetranucleotide usage patterns in DNA | [93] |
| METAREP | Web server | http://www.jcvi.org/metarep/ | Flexible comparative metagenomics framework | [98] |
| CD-HIT | Web server | http://weizhonglab.ucsd.edu/cd-hit/ | Identity-based clustering of sequences | [99] |
| METAGENassist | Web server | http://www.metagenassist.ca/ | Performs comprehensive multivariate statistical analyses on the data from different host and environment sites | [100] |
| CoMet | Web server | http://comet.gobics.de/ | ORF finding and subsequent Pfam domain assignment to protein sequences | [101] |
| WebCARMA | Web server | http://webcarma.cebitec.uni-bielefeld.de/ | Unassembled reads as short as 35 bp can be used for the taxonomic classification with less false positive prediction | [102] |
| MG-RAST | Web server | https://metagenomics.anl.gov/ | High-throughput pipeline for functional metagenomic analysis | [103] |
| CAMERA | Web server | https://portal.camera.calit2.net/gridsphere/gridsphere | Provides list of workflows for WGS data analysis | [104] |
| WebMGA | Web server | http://weizhonglilab.org/metagenomic-analysis/ | Implemented to run in parallel on local computer cluster | [105] |

# PICRUST

## Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences

Morgan G I Langille[1,14], Jesse Zaneveld[2,14], J Gregory Caporaso[3,4], Daniel McDonald[5,6], Dan Knights[7,8], Joshua A Reyes[9], Jose C Clemente[10], Deron E Burkepile[11], Rebecca L Vega Thurber[2], Rob Knight[10,12], Robert G Beiko[1] & Curtis Huttenhower[9,13]

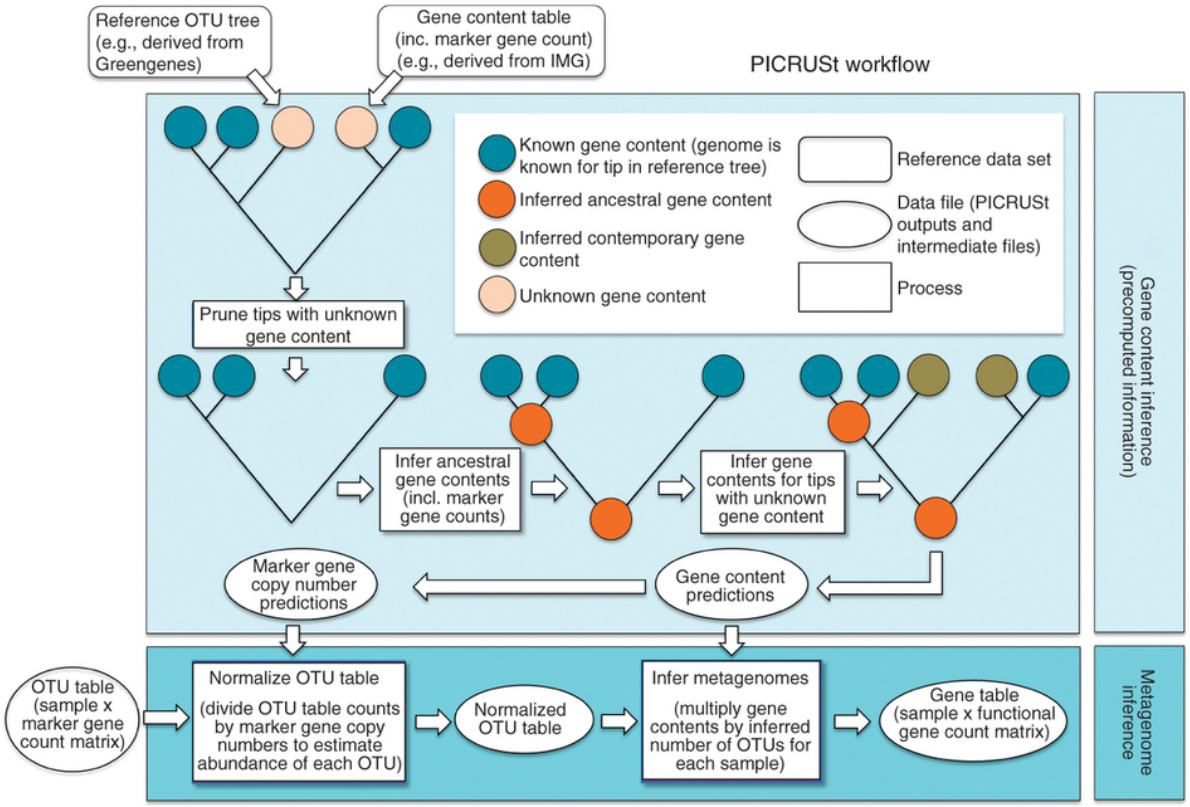
**Profiling phylogenetic marker genes, such as the 16S rRNA gene, is a key tool for studies of microbial communities but does not provide direct evidence of a community's functional capabilities. Here we describe PICRUSt (phylogenetic investigation of communities by reconstruction of unobserved states), a computational approach to predict the functional composition of a metagenome using marker gene data and a database of reference genomes. PICRUSt uses an extended ancestral-state reconstruction algorithm to predict which gene families are present and then combines gene families to estimate the composite metagenome. Using 16S information, PICRUSt recaptures key findings from the Human Microbiome Project and accurately predicts the abundance of gene families in host-associated and environmental communities, with quantifiable uncertainty. Our results demonstrate that phylogeny and function are sufficiently linked that this 'predictive metagenomic' approach should provide useful insights into the thousands of uncultivated microbial communities for which only marker gene surveys are currently available.**



phylogenetic diversity of environmental samples. Because marker gene studies focus on one or a few universal genes, they cannot directly identify metabolic or other functional capabilities of the microorganisms under study. Conversely, metagenomic sequencing aims to sample all genes from a community and can produce detailed metabolic and functional profiles. Although relatively little sequencing is needed to characterize the diversity of a sample[3,4], deep, and therefore costly, metagenomic sequencing is required to access rare organisms and genes[5]. Thus, marker gene profiling of large sample collections is now routine, but deep metagenomic sequencing across many samples is prohibitively expensive.

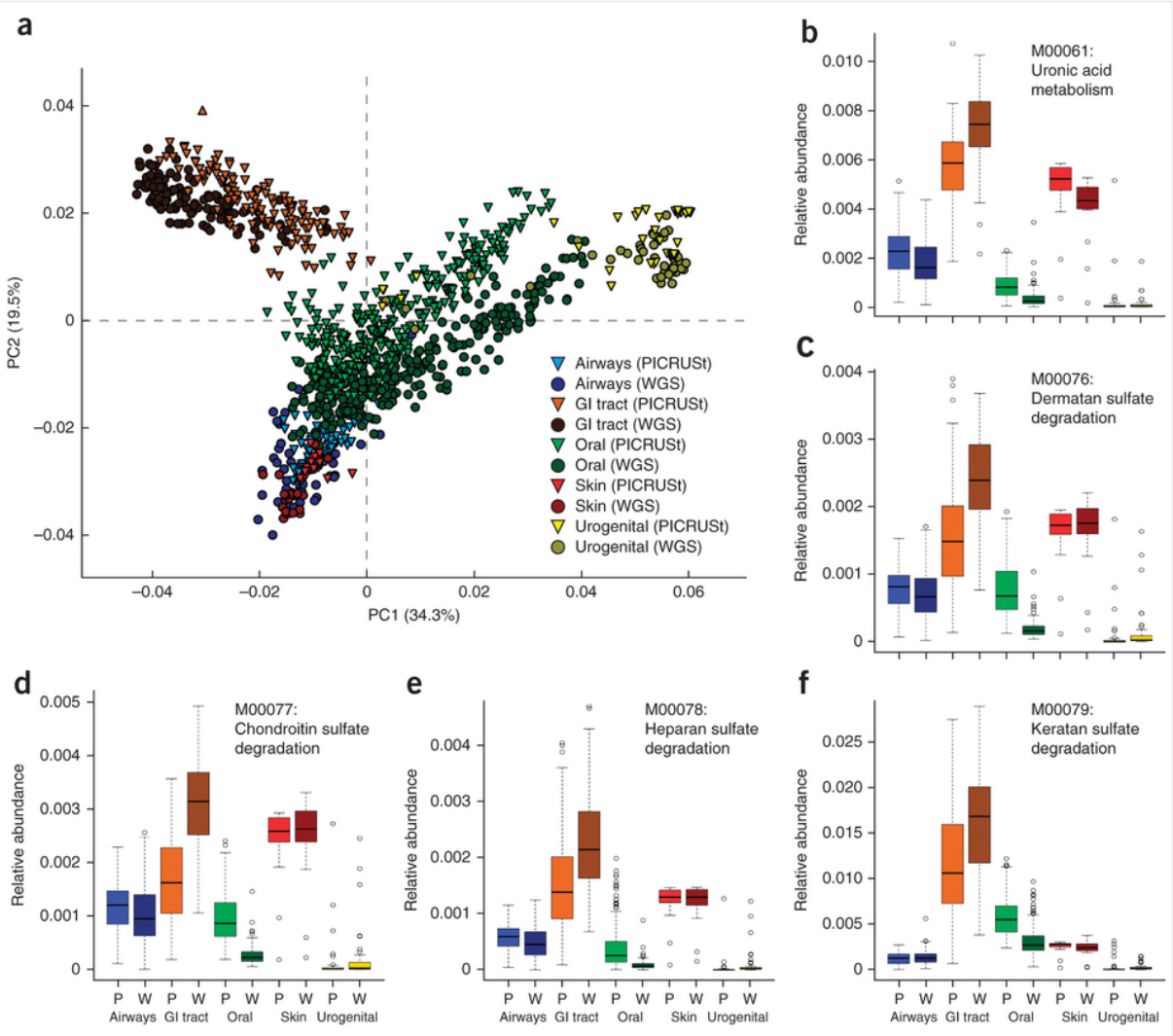
Although marker gene and shotgun sequencing strategies differ in the type of information produced, phylogeny and biomolecular function are strongly, if imperfectly, correlated. Phylogenetic trees based on 16S closely resemble clusters obtained on the basis of shared gene content[6–9], and researchers often infer properties of uncultured organisms from cultured relatives. For example, the genome of a *Bacteroides* spp. might reasonably be inferred to contain many genes encoding glycoside hydrolase activity, based on the commonality of these activities in sequenced *Bacteroides* isolates[10]. This associa-

# PICRUST



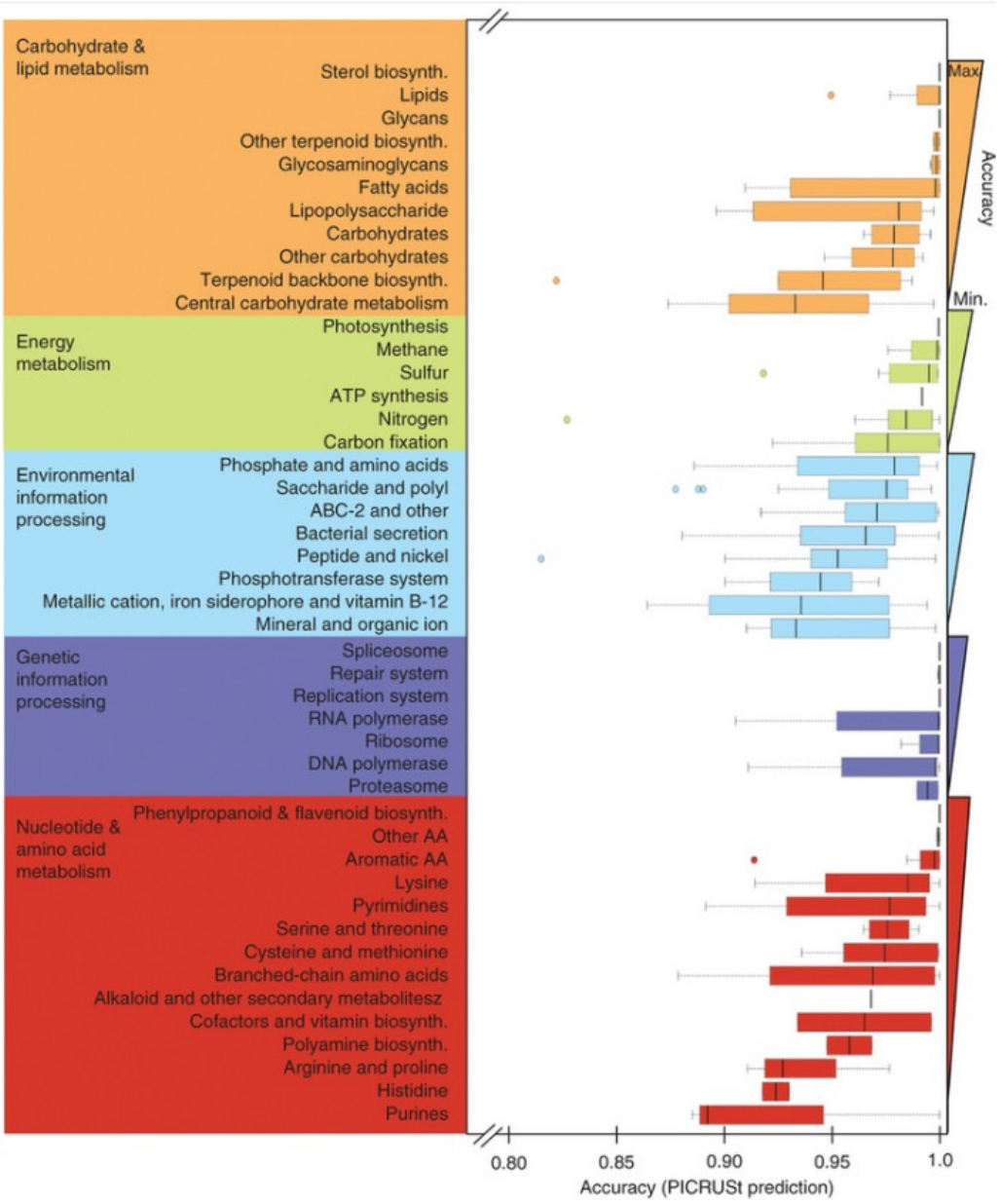

PICRUSt is composed of two high-level workflows: gene content inference (top box) and metagenome inference (bottom box). Beginning with a reference OTU tree and a gene content table (i.e., counts of genes for reference OTUs with known gene content), the gene content inference workflow predicts gene content for each OTU with unknown gene content, including predictions of marker gene copy number. This information is precomputed for 16S based on Greengenes[29] and IMG[26], but all functionality is accessible in PICRUSt for use with other marker genes and reference genomes. The metagenome inference workflow takes an OTU table (i.e., counts of OTUs on a per sample basis), where OTU identifiers correspond to tips in the reference OTU tree, as well as the copy number of the marker gene in each OTU and the gene content of each OTU (as generated by the gene content inference workflow), and outputs a metagenome table (i.e., counts of gene families on a per-sample basis).

# PICRUST




(**a**) Principal component analysis (PCA) plot comparing KEGG module predictions using 16S data with PICRUSt (lighter colored triangles) and sequenced shotgun metagenome (darker colored circles) along with relative abundances for five specific KEGG modules: (**b**) M00061: Uronic acid metabolism. (**c**) M00076: Dermatan sulfate degradation. (**d**) M00077: Chondroitin sulfate degradation. (**e**) M00078: Heparan sulfate degradation. (**f**) M00079: Keratan sulfate degradation. All KEGG modules are involved in glycosaminosglycan degradation (KEGG pathway ko00531) using 16S with PICRUSt (P) and whole genome sequencing (W) across human body sites. Color key is the same as in **a**.

# PICRUST




Results are colored by functional category and sorted in decreasing order of accuracy within each category (indicated by triangular bars, right margin). Note that accuracy was >0.80 for all, and therefore the region 0.80–1.0 is displayed for clearer visualization of differences between modules.

# PICRUST

http://picrust.github.io/picrust/



## PICRUSt: Phylogenetic Investigation of Communities by Reconstruction of Unobserved States

PICRUSt (pronounced "pie crust") is a bioinformatics software package designed to predict metagenome functional content from marker gene (e.g., 16S rRNA) surveys and full genomes.

PICRUSt is freely available under the GPL.

### Using PICRUSt

If you're new to PICRUSt, you'll want to work through these documents in order:

1. *Installing PICRUSt* **OR** Use online Galaxy version.
2. *Quickstart Guide*
3. *Metagenome Prediction Tutorial*
4. *Analyzing PICRUSt predicted metagenomes*
5. *Quality Control of PICRUST Predictions*

More advanced users may be interested in the following (in no particular order):

- *How PICRUSt Works*
- *Genome Prediction Tutorial*
- Installing PICRUSt in Galaxy
- *PICRUSt Script Index*
- *PICRUSt Methods*

### Contact

For PICRUSt announcements and questions, including notification of new releases, you should subscribe to the PICRUSt users list.

### Citing PICRUSt

On this page we've compiled both the PICRUSt citation and links to several tools that PICRUSt is built on to make citing these various software packages easier.

**Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences.** Langille, M. G.I.*; Zaneveld, J.*; Caporaso, J. G.; McDonald, D.; Knights, D.; a Reyes, J.; Clemente, J. C.; Burkepile, D. E.; Vega Thurber, R. L.; Knight, R.; Beiko, R. G.; and Huttenhower, C. *Nature Biotechnology*, 1–10. 8 2013.

The manuscript describing PICRUSt can be found here

- Additional citation resources: *here*

### News & Announcements

- 09-03-13: Official release of PICRUSt 1.0.0
- 08-25-13: PICRUSt published in Nature Biotechnology
- 07-10-13: PICRUSt presented by Morgan Langille at Gordon Research Conference: Applied & Environmental Micobiology
- More *News & Announcements*

# PICRUST

http://picrust.github.io/picrust/tutorials/qiime_tutorial.html#qiime-tutorial

## Analyzing metagenomes with QIIME

Because the metagenomes are provided in BIOM format by default, these can be plugged into many of the downstream analysis tools available in QIIME. QIIME's Shotgun Metagenome Analysis tutorial illustrates a couple of the steps that can be applied. The steps that will primarily be of interest in that tutorial are the ones that begin with a `.biom` file. For example, computing beta diversity and PCoA plots and generating summaries of samples by KO categories.

Many of QIIME's tutorials that describe diversity analyses are applicable to PICRUSt-predicted metagenome tables. Specific analysis tools that may be useful include:

- alpha_diversity.py
- beta_diversity.py
- compute_core_microbiome.py
- jackknifed_beta_diversity.py
- make_distance_boxplots.py
- alpha_rarefaction.py
- beta_diversity_through_plots.py
- group_significance.py
- shared_phylotypes.py

Plots of functional categories at various levels can be created using summarize_taxa_through_plots.py

- Since KEGG Orthologs belong to several pathways you should collapse your PICRUSt predictions to the desired hierarchy level using *categorize_by_function.py*

```
categorize_by_function.py -i metagenome_predictions.biom -c "KEGG_Pathways" -l 2 -o metagenome_at_level2.biom
```

- Then add the following lines to a qiime parameter file (e.g. qiime_params.txt) ensuring that the level you collapsed at is the same in your config file

```
summarize_taxa:md_identifier    "KEGG_Pathways"
summarize_taxa:absolute_abundance    True
summarize_taxa:level    2
```

- Lastly, run summarize_taxa_through_plots.py

```
summarize_taxa_through_plots.py -i metagenome_at_level2.biom -p qiime_params.txt -o plots_at_level2
```

There are also a number of scripts in QIIME that may be useful for more general processing of your BIOM table. These include the following:

- single_rarefaction.py
- filter_otus_from_otu_table.py
- filter_samples_from_otu_table.py
- per_library_stats.py
- filter_taxa_from_otu_table.py
- merge_otu_tables.py
- sort_otu_table.py
- split_otu_table.py
- split_otu_table_by_taxonomy.py

Note that while many of these refer to OTU table, it's just a nomenclature issue. These are generally applicable to `.biom` tables.

Finally, if you're interested in comparing real to predicted metagenomes, or predicted metagenomes to 16S, you'll be interested in the Procrustes Analysis tutorial and the Comparing Distance Matrices tutorial.

# PRMT

## Predicted Relative Metabolomic Turnover (PRMT): determining metabolic turnover from a coastal marine metagenomic dataset

Peter E. Larsen[1*], Frank R. Collart[1], Dawn Field[2], Folker Meyer[1], Kevin P. Keegan[1], Christopher S. Henry[1,4],John McGrath[3], John Quinn[3], Jack A. Gilbert[1,5]

[1] Argonne National Laboratory, 9700, S. Cass Ave, Argonne, Illinois, USA
[2] NERC Centre for Ecology and Hydrology, CEH Oxford, Mansfield Road, Oxford, OX1 3SR, UK
[3] School of Biological Science, Queens University, Medical Biology Centre, 97 Lisburn Road. Belfast, BT9 7BL, Northern Ireland, UK
[4] Computation Institute, University of Chicago, 5735 South Ellis Avenue, Chicago, IL 60637, U.S.A
[5] Department of Ecology and Evolution, University of Chicago, 5640 South Ellis Avenue, Chicago, IL 60637, U.S.A

* Corresponding author

Email PEL: plarsen@anl.gov
FRC: fcollart@anl.goc
DF: dfield@ceh.ac.uk
FM: folker@mcs.anl.gov
KPK: kkeegan@anl.gov
CSH: chrisshenry@gmail.com
JM: j.mcgrath@qub.ac.uk
JQ: j.quinn@qub.ac.uk
JAG: gilbertjack@gmail.com

# PRMT

**PRMT analysis approach**

PRMT scores predict the change in turnover of metabolites (defined as the potential for consumption or production) in an environmental metabolome, given the relative abundance of genes for unique enzyme functions detected in different metagenomes. In this manuscript, we use the term "unique enzyme function" to describe a specific annotation applied to an enzyme, i.e. "Phosphotransferases with an alcohol group as acceptor". We use "enzyme reactions" to refer to metabolite transformations catalyzed by an enzyme function, i.e "ATP + D-Glycerate ↔ ADP + 3-Phospho-D-glycerate". A unique enzyme function may catalyze more than one enzyme reaction and an enzyme reaction may be catalyzed by more than one unique enzyme function. A metabolite is a molecular compound that is a reactant or product in an enzyme reaction. In PRMT, a metabolite is never the protein product of a gene in the metagenome.

# PRMT

**A**

$C1 \xrightarrow{a} C2$

$C1 \xrightarrow{a} C3$

$C3 \xrightarrow{b} C1$

$C2 \xrightarrow{c} C3$

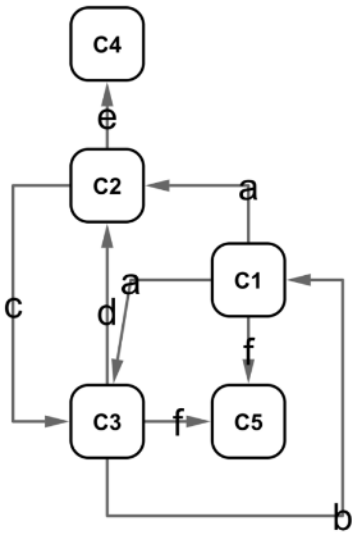
$C3 \xrightarrow{d} C2$

$C2 \xrightarrow{e} C4$

$C1 \xrightarrow{f} C5$

$C3 \xrightarrow{f} C5$

**B**




**C**

| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ |
|---|---|---|---|---|---|---|
| **C1** | -2 | 1 | 0 | 0 | 0 | -1 |
| **C2** | 1 | 0 | -1 | 1 | -1 | 0 |
| **C3** | 1 | -1 | 1 | -1 | 0 | -1 |
| **C4** | 0 | 0 | 0 | 0 | 1 | 0 |
| **C5** | 0 | 0 | 0 | 0 | 0 | 2 |

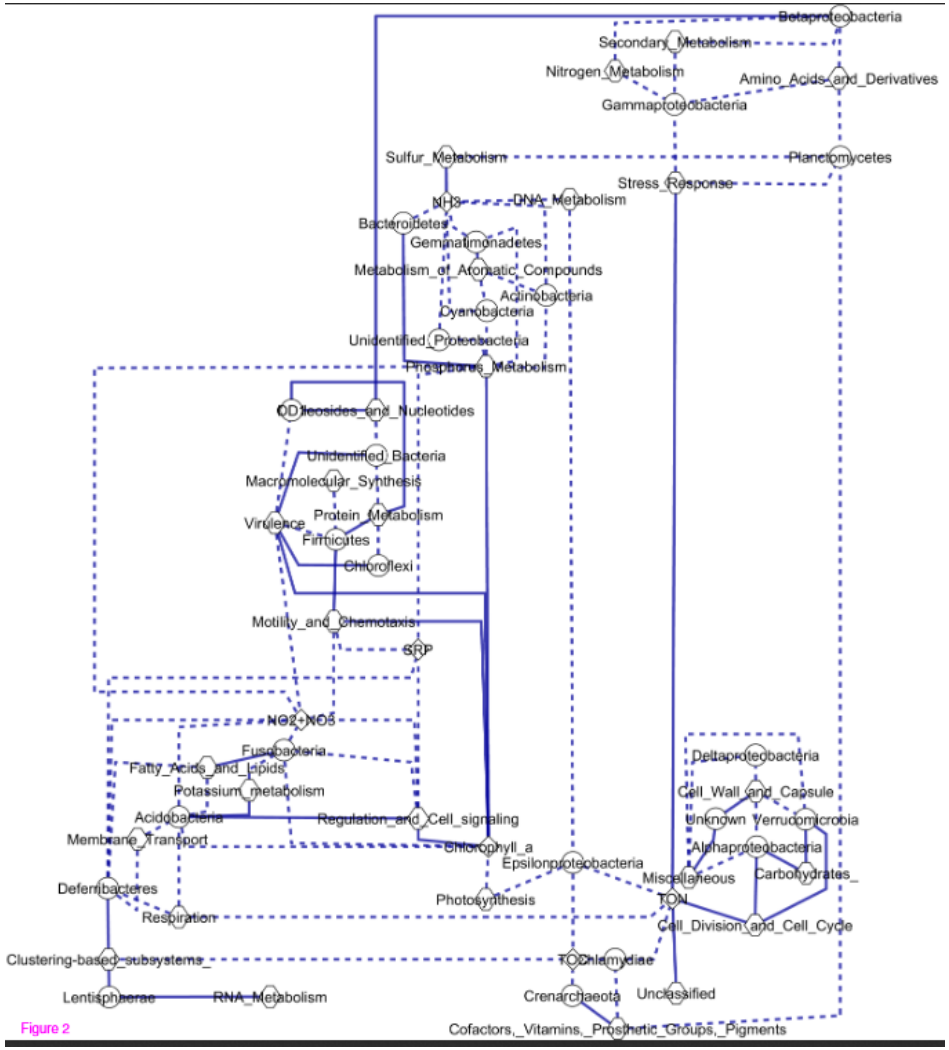
**D**

| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ |
|---|---|---|---|---|---|---|
| **C1** | -0.66 | 1 | 0 | 0 | 0 | -0.33 |
| **C2** | 0.5 | 0 | -0.5 | 0.5 | -0.5 | 0 |
| **C3** | 0.5 | -0.33 | 0.5 | -0.33 | 0 | -0.33 |
| **C4** | 0 | 0 | 0 | 0 | 1 | 0 |
| **C5** | 0 | 0 | 0 | 0 | 0 | 1 |

Figure 1

Example of generating an EMM from metagenomic data. This figure is an example of generating a simple EMM with hypothetical data. Letters a-f represent unique enzyme functions identified in the annotation of a hypothetical set of metagenomes. In (A), the set of all enzyme reactions for enzyme functions a-f between compounds C1-C5 from a database of possible reactions is listed. In (B), a metabolome is constructed from the reactions identified in A. (C) Shows the connectivity matrix of the network in B. (D) Is the complete EMM for metagenomic annotated enzyme functions a-f, normalizing values in C such that the sum of all inputs to a compound is 1 and the sum of all outputs from a compound is -1.

# PRMT




Strong correlations between environmental metabolites, metabolic subsystems, and bacterial population structure.
This network is a graphical representation of strong (i.e. in the top or bottom 5th percentile of randomized resamples) correlations between relative abundance of measured environmental metabolites (diamonds), relative abundance of metagenomic reads annotated to metagenomic SEED subsystems (hexagons), and relative abundance of bacterial taxa (circles) across seasonal variation for the Western English Channel L4 station. Strong positive correlations are represented by solid lines and strong negative correlations by dashed lines.

# PRMT

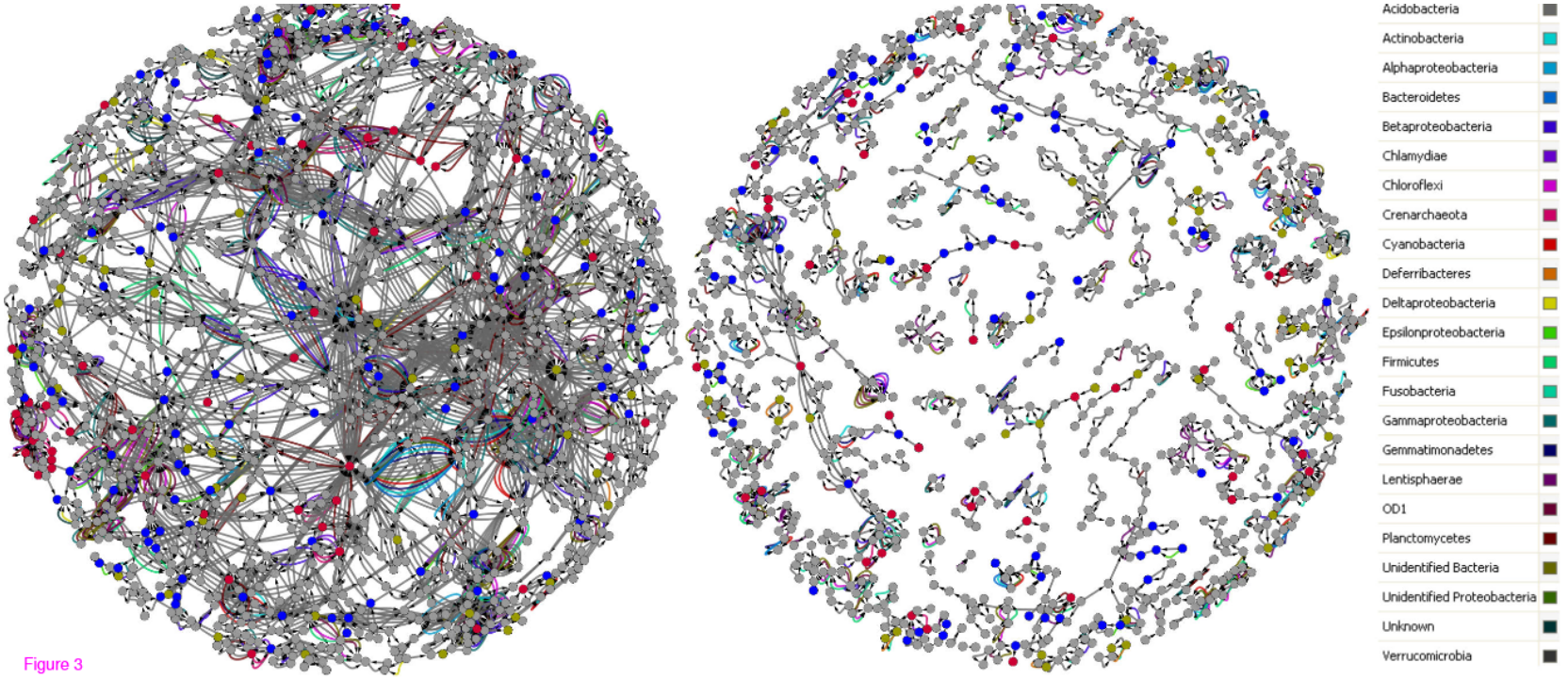


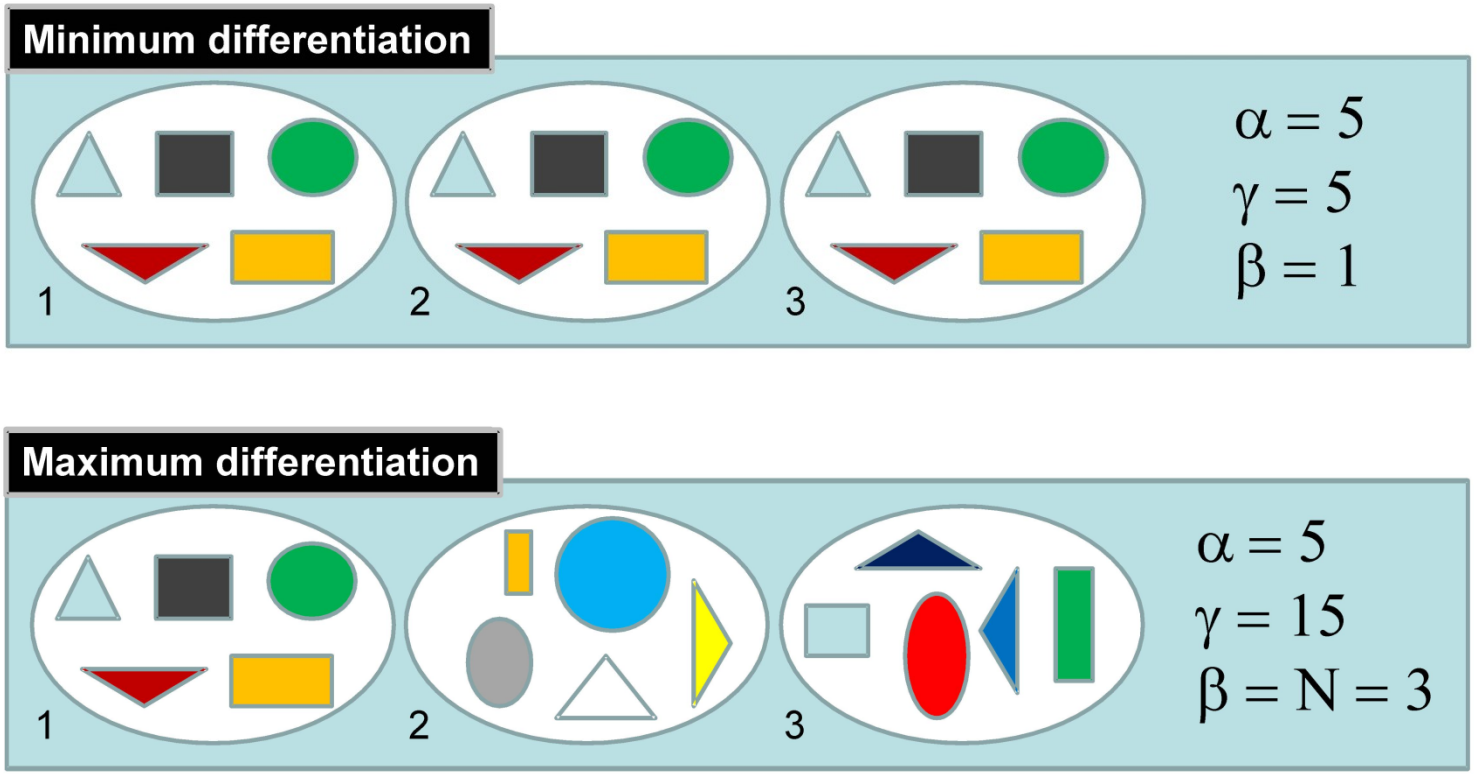

Figure 3

L4 Environmental Metabolome

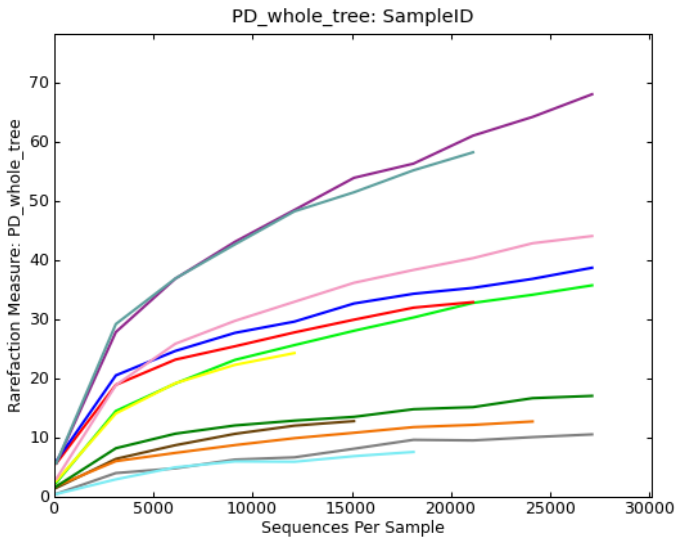
In the figure, edges represent enzyme functions identified in annotated metagenomes. Nodes are predicted metabolites, inferred by the reactions catalyzed by detected enzyme functions. Nodes are highlighted if calculated PRMT scores for seasonal metagenomes correlate strongly (i.e. in the top or bottom 5$^{th}$ percentile of randomized resamples) with relative abundance of measured environmental parameters (Red for Total Organic Carbon, blue for Total Organic Nitrogen, and gold for Soluble Reactive Phosphorus). Edges are highlighted in one of 23 colors if they connect nodes that correlate with relative abundance of a bacterial phylum. Figur e was generated using Cytoscape v2.6.1. The network and calculated PRMT-scores in this figure is available for download as additional file 3, figure S1.

# Diverzita

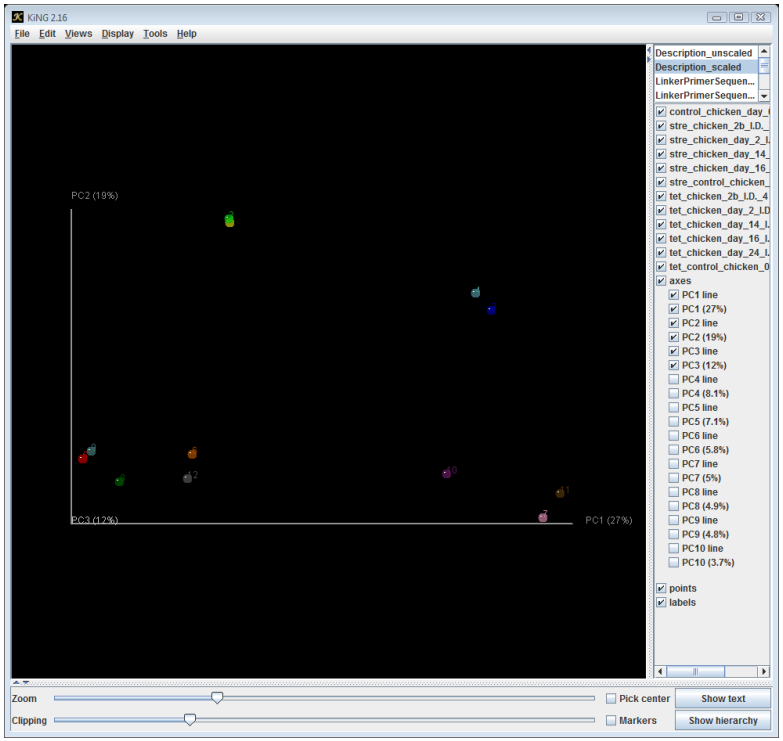
- $\alpha$ vs $\beta$ diverzita
- https://methodsblog.wordpress.com/2015/05/27/beta_diversity/



# α vs β diverzita




vs.



# Indexy $\alpha$ diverzity

- **Shannon index** – započítává vyrovnanost (eveness) i abundanci druhů vyskytujících se ve vzorku

$$H = -\sum_{i=1}^{k} p_i \log p_i$$

- **Simpson's index** – zvažuje výskyt nejvíce zastoupených druhů → měří pravděpodobnost, že dvě náhodně vybraná individua budou patřit do stejného druhu

$$D = 1 - \left( \frac{\Sigma\, n(n-1)}{N(N-1)} \right)$$

# Indexy $\alpha$ diverzity

- **Chao1 estimator** – odhaduje pravdivou druhovou diverzitu ve vzorku

$$S_1 = S_{obs} + \frac{F_1^2}{2F_2}$$

http://palaeo-electronica.org/2011_1/238/estimate.htm