Note - original files for nemdb3 are in:

/data/seq_tables/ - real format

- BLAST databases are in /etc/ncbi/db

Get Species

1. Create directory est_solutions.
2. Genereate nematode EST source page from NCBI:
   - Search Taxonomy for Nematoda
   - Display 10 levels
   - Nucleotide EST
3. Run www.taxparser.pl on source (note would be much better to just output fasta files for existing species as separate seq files aren't necessary for part 9.
4. Run download_txdlist2.pl on output of 3 within est_solutions directory

#### note this seems to be outputing more files than EST number on NCBI, might have to run Partigene for each species as this appears to download correct number of sequences - problem appears to be that download_txdlist2.pl downloads mRNA as well as ESTs, just need to edit line 29 ###

PartiGene

5. For new species run Partigene.pl within species directory - split seqs up and put in sequences directory using
   split_multiple_contig_fasta_into_single_fasta.pl
6. For species already present in nemdb run psql2txt.pl on nemdb to generate 'XXX'EST files for each species using the following sql command.
   - perl psql2txt.pl nemdb3 "select est.est_id ||' '|| clus_id,sequence from est,est_seq where clus_id ~ 'XXX' and est.est_id = est_seq.est_id" XXXEST fasta;
   - Note, need to be logged in as postgres but within home directory when running on xyala
7. Place each EST file in appropriate species directory.
8. Create hash of nemdb EST identifiers.
9. Run ncbi_vs_nemdb_bioperl.pl which removes sequences from each species sequence folder and adds only new ones.
10. If there are no new species then ncbi_vs_nemdb wont work as it will leave the sequences folder empty. In this case run Partigene_restart.pl in PartiGene folder
    - NOTE!!!!! if species ESTS ans Clusters already generated and added to database, then if you want to replace them, then you have to delete both est and est_seq tables!!!!!
11. Run Partigene.pl on updated species, starting at Pre-process sequences step.
12. Add all info to new database.

BLAST

12. Run first step on Partigene BLAST to produce BLAST .txt output file
13. Run this on eddie and store results in named blast directories (e.g. uniref100) within blast directory for each species (run shell script to generate these)

```
        sge_blastall -i blast_input/blast_input_AYC.txt -d
/exports/work/blast/uniref100.fasta -o blast_output/AYC_uniref.out -p blastx -b 5 -v 5 -e
1e-8 -N 500; sleep 3;
```

- NOTE, if blast file is concatenated run blast_to_many.pl specifying parameters,

e.g. **blast_to_many.pl -d wormpep/ -s "out" wormpep/ZPC_wormpep.out**

14a. Store results in DB

14b. Note, hit value is not present, add column to blast table then add values using pg_blasthit.pl (**NOTE!!!** - need to install Pg.pm -> sudo apt-get install libpg-perl)

Create and fill blast_top table using create_and_fill_blast_top.pl

prot4EST

15. Set config file to use protein directory and BLAST results for each species (prot4EST_config_maker.pl)
    - if doesnt work try with another species by manually changing the config file

15.b If using new p4e results, make sure IDs use P not C, as all scripts and such expect it, i.e. ABP00012_1
   - use change_clusterID_to_proteinID.pl
   - insert protein seqs into db using newer_p4e_protein_parser.pl

16. Store results in DB

annot8r

17. Take all protein predictions from prot4EST and BLAST against uniprotEC/KEGG/GO then run annot8r.pl to update to DB.
    - don't BLAST as m8!!!!!!!!!!!!!!!
    - to create dbs need to be logged in as postgres user - createdb a8r_gobase -O ben
    - change use of redirect '>& /dev/null' to '> /dev/null 2>&1'
    - change charvar(12) as not long enough any more
    - see annot8r_blast.sh (or blast_db_maker.sh to generate single cat file of all peptides)
    - get blast dbs by running annot8r.pl, downloading data and creating BLAST dbs
    - when running on eddie use this:

```
sge_blastall -i annot8r_input.fsa -d
/exports/work/biology_ieb_mblaxter/uniprot_KEGG.fsa -o annot8r_output/annot8r_KEGG.out
-p blastp -b 1 -v 1 -e 1e-08 -N 500;
```

    - note, a8r_gobase needs tables altering, changed length of char vat(12) to (50)

18**. REMEMBER!!!!!** Put all blast output (even from blast farm) in blast_out within directory you are running annot8r.pl from otherwise it will not work!!!!
    - Also, doesn't seem to like running all together, do one at a time or fix it!
    - Parameters used - evalue, 1e-08, 1 hit

NemDom

- pfam results given by James in Jan 09

- <seq_id> <seq_start> <seq_end> <hmm_acc> <hmm_start> <hmm_end> <bit_score> <evalue> <hmm_name>

e.g.

```
O00519 95 562 PF01425.9 1 513 526.5 2.6e-155 Amidase
O01636 22 139 PF02408.8 1 137 157.3 3.6e-44 DUF141
O03046 12 137 PF02788.5 1 132 295.2 1.1e-85 RuBisCO_large_N

see http://pfam.sanger.ac.uk/help

Parse and add to db?
```

NemPep

19. Is the complete set of protein predictions generated from prot4EST

NemFam

20. Blast protein database against itself (fasta file against created DB) and use this as input for MCL (Tribe).
    - take all the peptide seqs predicted through p4e, make into fasta file and then formatdb it
    - blastp this file against the database of itself
    - blastall -i prot4EST_output.fsa -d nempep -o nempep_prac -p blastp -b 20 -v 20 -e 1e-08;

Interpro

21. A tool that combines different protein signature recognition methods native to the InterPro member databases into one resource with look up of corresponding InterPro and GO annotation.
    - setup, run sudo perl Config.pl
    - set chunk size to 1000 to avoid blowing up processors!!!
    - feed in protein predictions and parse through output and update to db.
    - can run on normal pc for small data: iprscan -cli -i prot4EST_output/translations_xtn.fsa -iprlookup -goterms -f raw > interproscan.out

- create interpro_key table using short_names.dat and protein2ipr.dat from interpro ftp using interpro_key_db.pl
- create interpro table using interproscan_parser.pl

Interpro on eddie

    - should really be run on eddie, but note that it requires 22gb of space - remember to run with qsub at front of command to put job on separate node,
        - need to use specific script (see sge_interproscan.pl) to generate shell script!!! NB..see also sge_maq.pl
            *** was missing some perl modules so got eddie people to install, but script needs to add these to PERL5LIB (export PERL5LIB=/exports/work/apps/perl/lib/perl5) each time as just having them locally wont work!!! ***
        - setting output file name (> test.out) is necessary otherwise it crashes
        - f chooses output format to be tab-delim
    - to add to db use interproscan_parser.pl

Was having problems with indexing, especially of match_complete.xml, to re-index:

```
        perl bin/index_data.pl -inx -iforce -v
```

see ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/FAQs.html

Getting bad fd number error, suggesting issues with v4.4 and ubuntu, had to construct the binaries using -bin in index_data.pl on eddie and transfer across!

see line 342:

```
            if(system("$path/bin/binaries/hmmconvert -b $f $f.bin >& /dev/null")){
```

changed it to >/dev/null 2>&1

NOTE! - possible error, see below! (http://sm.life.nthu.edu.tw/doc/FAQs.html#26)

InterProScan gives me a report file containing
some errors from FingerPRINTScan that are weird
like : ERROR: Calculation has exceeded maximum allowed complexity
Fingerprint PRICHEXTENSN matches this sequence
==

```
This is not a real error, this is just a warning. Don't worry about it.
```

- was still not finding the perl modules, so copied directory over to one in perl path
- cp -r /exports/work/apps/perl/lib/perl5/* /exports/work/biology_ieb_mblaxter/ben/iprscan_ben/iprscan/lib

1. Set up conf (perl Config.pl) - chunk size big enough to produce less than 5000 jobs as eddie can't handle more than that. Use sge queue sustem, don't set names. Change max number of sequences permitted to larger number (e.g. 1000000).

2. Set shell script to run initial command, i.e. standard iprscan command using full pathways, e.g.

```
#!/bin/sh
# Grid Engine options
#$ -N sge_iprscan
#$ -cwd
# Initialise environment module
. /etc/profile.d/modules.sh
# Use Intel compiler
module load intel/icc/64
#set environment variables
export PERL5LIB=/exports/work/apps/perl/lib/perl5
# Run the program
/exports/home/s9901625/bin/iprscan -cli -i
/exports/work/biology_ieb_mblaxter/ben/iprscan_ben/nempep/AYC.out -iprlookup -goterms
-format raw -o /exports/work/biology_ieb_mblax
ter/ben/iprscan_ben/iprscan/iprscan.out
```

Note - no two space gaps in command line.
- can change parameters by adding things to conf/sge6.conf , like -P belswort_01- setting parallel nodes....in theory!

and then run:

```
qsub iprscan.sh
```

Note - make sure the env variable has been loaded, otherwise this will not work and will just stall!

#######Dont even need to do this, just run iprscan command as normal, it will fail but create the .dcmd dile in tmp and then move to the next step

3. This fails (not sure why, gives error of not recognising qsub command), so manually run all the new .dcmd files in the chunks

```
tmp/.../.../.../.../...dcmd
```

This creates all the necessary output files but doesn't collate them together.

4. perl bin/out2raw.pl -j *job_id* -o *xxx.out* -v

To check how many jobs are running use:

```
qstat | grep -c 's9901625'
```

sex_count and stage_count tables

22. First if necessary update the lib table with new stage names, see libs2.cvs and use add_marks_lib_info.pl

Create sex_count and stage_count tables using appropriate psql

```
create table lib_key(lib_id text not null,species text,spec_id text,sex text,stage text);

insert into lib_key(lib_id,species,sex,stage) select lib_id,organism,sex,stage from lib;

update lib_key set spec_id = species.spec_id from species where
lib_key.species=species.species;

CREATE TABLE sex_count (
    clus_id character varying(10),
    female integer,
    male integer,
    mixed integer,
    total_ests integer
);

CREATE TABLE stage_count (
    clus_id character varying(10),
    adult integer,
    eggs integer,
    l1 integer,
    l2 integer,
    l3 integer,
    l4 integer,
    mixed integer,
    unknown integer,
    total_ests integer,
    adult_p integer,
    eggs_p integer,
    l1_p integer,
    l2_p integer,
    l3_p integer,
    l4_p integer,
    mixed_p integer,
    unknown_p integer,
    spid character varying(5)
);
```

-Run sex_stage_count_table_builder.pl or stage_count_table_builder.pl to populate sex_count and stage_count tables using lib data

Update table with proportions using sql commands:

```
psql>update stage_count set eggs_p = cast(eggs as real) / total_ests * 100;
```

Nembase Frontend

23. Update all scripts to coincide with changes and developments of PartiGene, prot4EST, annot8r....
24. Update species_db by copying over species table from nemdb4 and updating using relevant commands.
    - run add_directories_and_file_to_species_db.pl to update directory name to match species name
25. Current scripts require info from signalp and psort
    - install local version of signalp
    - run signalp_for_each_species.sh
    - database using signalp_db.pl
26. Need to setup dbs to run with webuser:

```
        psql <databasename>
grant select on a8r_blastec, a8r_blastgo, a8r_blastkegg, blast,blast_top, clone_name, cluster, ec2description, est, est_seq, genome_pep, hit_table, interpro, interpro_key, lib, lib_count, lib_key, node2tribe, node_stats, p4e_hsp, p4e_ind, p4e_loc, pathway_id2name, pathway_map, reciprocals, sex_count, signalp, species, stage_count, tribe, tribe_info, tribe_node to webuser;
```

signalp

27. Edit signalp script to set up necessary variables
28. If necessary, run split_fasta_by_set_number_of_seqs.pl to split seqs into chunks of 1000 or so, as signalp doesn't like big files
28. Check all directories are setup and then run signalp_for_each_species2.sh
29. Run signalp_db.pl

Reactome / pathway stuff

30. Added clade info to species table - made no real difference
31. \o cladeV.out

```
select distinct(ec_id) from a8r_blastec where pept_id ~ 'HCP' or pept_id ~ 'TDP' or pept_id ~ 'NAP' or pept_id ~ 'NBP' or pept_id ~ 'PPP' or pept_id ~ 'AAP' or pept_id ~ 'ABP' or pept_id ~ 'ACP' or pept_id ~ 'AYP' or pept_id ~ 'OOP' or pept_id ~ 'CBP' or pept_id ~ 'CGP' or pept_id ~ 'CJP' or pept_id ~ 'CRP' or pept_id ~ 'CSP' or pept_id ~ 'HBP';

\o cladeIV.out
```

select distinct(ec_id) from a8r_blastec where pept_id ~ 'SRP' or pept_id ~ 'SSP' or pept_id ~ 'GRP' or pept_id ~ 'MJP' or pept_id ~ 'MIP' or pept_id ~ 'ZPP' or pept_id ~ 'DVP' or pept_id ~ 'HSP' or pept_id ~ 'ODP' or pept_id ~ 'PAP' or pept_id ~ 'PTP' or pept_id ~ 'BUP' or pept_id ~ 'BXP' or pept_id ~ 'DAP' or pept_id ~ 'GMP' or pept_id ~ 'GPP' or pept_id ~ 'HGP' or pept_id ~ 'PEP' or pept_id ~ 'PVP' or pept_id ~ 'RSP' or pept_id ~ 'MAP' or pept_id ~ 'MCP' or pept_id ~ 'MHP' or pept_id ~ 'MPP' or pept_id ~ 'PSP' or pept_id ~ 'SCP' or pept_id ~ 'SFP' or pept_id ~ 'TIP';
\o cladeIII.out
select distinct(ec_id) from a8r_blastec where pept_id ~ 'ASP' or pept_id ~ 'ALP' or pept_id ~ 'TCP' or pept_id ~ 'BMP' or pept_id ~ 'WBP' or pept_id ~ 'OVP' or pept_id ~ 'LSP' or pept_id ~ 'DIP' or pept_id ~ 'AIP' or pept_id ~ 'TLP' or pept_id ~ 'BPP' or pept_id ~ 'LLP' or pept_id ~ 'OCP' or pept_id ~ 'OFP';

\o cladeI.out
select distinct(ec_id) from a8r_blastec where pept_id ~ 'TSP' or pept_id ~ 'TMP' or pept_id ~ 'XIP' or pept_id ~ 'TVP';

32. Edit out header and spaces, then add info to SkyPainter (part of Reactome - EBI).
33. Compare across clades

Reciprocal BLASTs

Run change cluster_id_to_protein_id if fasta header are XXC and not XXP!!

34. Split p4e file using split_p4e_cat_file_into_species_files.pl then formatdb all protein predictions
    - process fasta headers for formatdb
    - recip_formatdb.sh
35. BLAST all against all
    - recip_blast.pl
36. Parse results and add to db
    - recip_blast_parse.pl

Creating nemdb_info

select avg(length(consensus)) from cluster where clus_id ~ 'ABC';

select avg(length(seq)) from p4e_ind where pept_id ~ 'ZPP';

select count(distinct(clus_id)) from blast where clus_id ~ 'ACC' and db = 'uniref100';
    - divide by number of peptides = % uniref hits

run nempep_percentage_blast_hits_calculator.pl on blast results for c.elegans and nempep

Setting up pfam

1. Download the HMMER2 software.
2. Download the Pfam files Pfam_ls, Pfam_fs, Pfam_ls.bin, Pfam_fs.bin, Pfam_ls.bin.ssi, Pfam_fs.bin.ssi, Pfam-A.seed and Pfam-C from the ftp site.
   These files contain the HMMs and additional information that is required to carry out the searches.
3. Download a copy of pfam_scan.pl from the ftp site.

This is a wrapper script around hmmpfam, the HMMER program that searches query sequences against a library of profile HMMs. If perldoc is installed, more detailed instructions on how to use pfam_scan.pl can be found by typing on the command line 'perldoc pfam_scan.pl'.

4. On the command line enter:pfam_scan.pl -d <directory_location_of_Pfam> <files fasta_file_of_proteome>
   For example if the files were downloaded into a folder called pfam_files, and the FASTA sequences were in a file called sequences.fasta, type:pfam_scan.pl -d pfam_files sequences.fasta
   pfam_scan.pl will search the FASTA sequences against the profile HMMs and report all matches to families that score higher than the manually set thresholds for each of the Pfam families. The output format will look something like this:<seq_id> <seq_start> <seq_end> <hmm_acc> <hmm_start> <hmm_end> <bit_score> <evalue> <hmm_name>
   e.g.
   ```
   O00519 95 562 PF01425.9 1 513 526.5 2.6e-155 Amidase
   O01636 22 139 PF02408.8 1 137 157.3 3.6e-44 DUF141
   O03046 12 137 PF02788.5 1 132 295.2 1.1e-85 RuBisCO_large_N
   ```

e.g. pfam_scan.pl -d /usr/local/genome/pfam/ -cpu 6 2_nematodes.out -o 2_nematodes_pfam.out

TribeMCL

http://www.micans.org/mcl/

1. cat all reciprocal BLASTs

2. run command below, varying I

```
cut -f 1,2,11 all_recip.txt  | mcl - --abc --abc-neg-log -abc-tf 'mul(0.4343), ceil(200)' -I 3.5
-o 3.5.out
```

3. add to databases using tribe_parser_db.pl, but if using non nematode data do BLAST step below first!!

Node Tribes

Generated data for node tribe tables using tribe_investigation_all.pl on tribe MCL data

Generated data for non nematode BLAST hits by BLASTing nempep4 against custom db

```
    ~/my_sge_blastall_no_group -i nempep4_all.fix -o nempep_and_genomic_non_nem_blast_out.txt -d
/exports/work/blast/genbank_non_nem_fix.fa -p blastp -e 1e-05 -b1 -v1 -N 80 -m8
```

Make sure tribe.non_nematode has default values of 0!!

Add all tribe and non nematode data using:
    tribe_parser_plus_non_nematode_db.pl

Generate date for node2tribe and tribe_node tables using:
tribe_invesigation_all_plus_genomic.pl

PHP scripts created to get this data and output accordingly

Adding genomic data

1. Download all genomic pepts from wormbase
2. BLAST against matching species using high evalue (e-65)
3. Run add_genomic_pepts_to_p4e_ind.pl to put new data in p4e_ind and genomepep tables
4. Run generate_fasta_from_p4e_ind.pl to create new fasta file of all proteins
5. Redo reciprocal BLASTs with this file
6. Create tribes from these using above tribe command
7. add to database using tribe_parser.pl (remove old ones first)
8. BLAST new pepts against non_nematode db on eddie
9. Add non_nematode BLAST info to tables using add_non_nematode_to_tribe1.pl scripts
10. Add tribe node data to node2tribe and tribe_node using tribe_investigation_all_plus_genomic.pl
11. Use new php pages which identify the three types of protein now present,
    i. nempep
    ii. nempep with genomic overlap (uses genomic sequence in place of EST and has link to wormbase)
    iii. genomic only (XXG identifier)
12. see below (fixing sql issues)

Fixing sql issues

Created node_stats table to contain data for tribe_tree_results.php
Populate using node_stats_table_builder.pl
Make sure count(distinct(tribe)) is used in the script or unique keys are set in node2tribe table as otherwise numbers can be wrong!

Creating new BLAST database

1. Tinker with blast.shtml to change databases and the like
2. Get all cluster files:

```
>find . -name "blast_input_*.txt" -exec cp {} all_clusters/ \;
```

3. Rename them using rename

```
> rename 's/\.txt$//' *
> rename 's/blast_input_//' *
```

4. Format them all using multiple_formatdb_clusters.pl

5. Make clade databases, cat and format

5. Copy files to blastdb of choice

Getting the cgi stuff to work

1. Changed the link on cluster.php to point to n4_align.cgi
2. n4_align.cgi now uses this path to find .ace files

/var/www/html/nembase4/nembase4_data/$organism/phrap/$cluster.ace
3. clus_img.php not points to the same directory store as above

Updating species_db4

1. Run update_species_db4_with_data.pl

Updating species table

1. Run update_species_table.pl

Pathways

1. Search using pathways.shtml
2. This points to pathway_results.php which calls nembase4_pathways.cgi.
3. Click on a pathway from the results goes to pathway_details.php which calls map.cgi with the necessary details.

Creating annotation files

Run create_description_fasta_files.pl