Neural-net Extracellular Trained Flux Balance Analysis, a hybrid approach to constrain genome-scale models

James Morrissey[1,3], Gianmarco Barberi[2,3], Benjamin Strain[1], Pierantonio Facco[2], Cleo Kontoravdi[1,4]

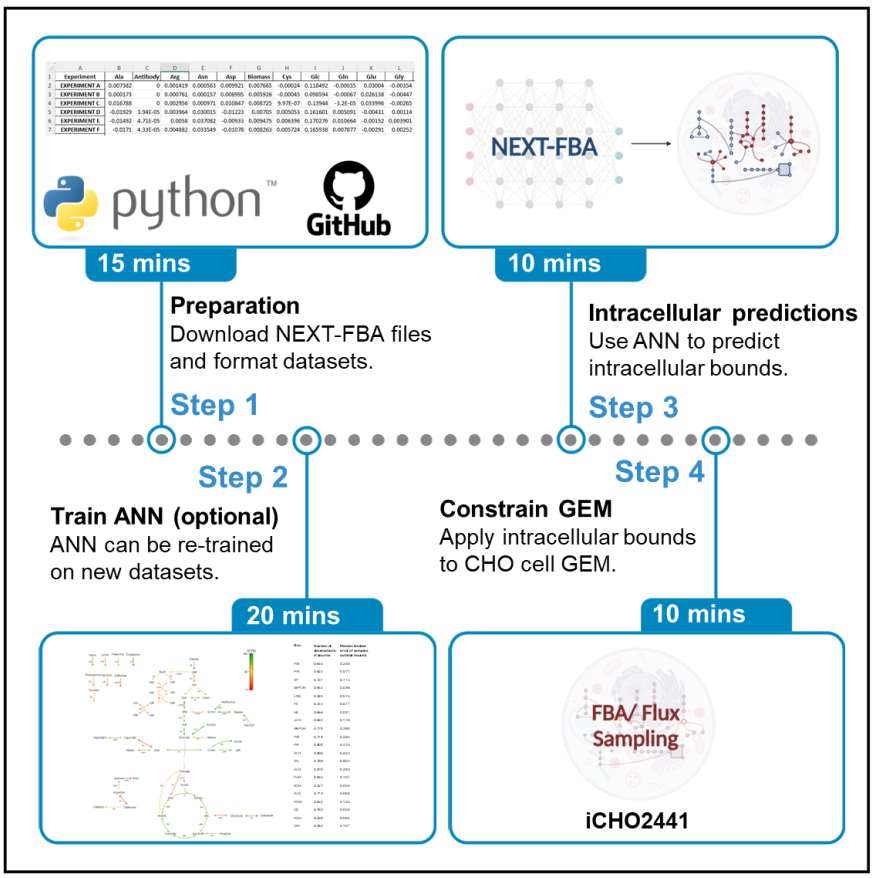
[1]Department of Chemical Engineering, Imperial College London, London, United Kingdom
[2]CAPE-Lab (Computer-Aided Process Engineering Laboratory), Department of Industrial Engineering, University of Padova, Padova, Italy

[3]Technical Contact: rjm216@ic.ac.uk, gianmarco.barberi@unipd.it
[4]Lead Contact: cleo.kontoravdi@imperial.ac.uk

Summary

NEXT-FBA is a hybrid mechanistic and data-driven model to constrain genome scale models (GEMs). NEXT-FBA involves training an artificial neural network (ANN) to understand the correlation between exometabolomics and cell metabolism. The ANN predicts intracellular reaction bounds for unseen data and these bounds are used to constrain a GEM. This protocol will walk through two separate procedures. Firstly, how to train the ANN on a new dataset. Secondly, applying a pre-trained ANN to constrain a Chinese hamster ovary (CHO) cell GEM.

Graphical abstract



Before you begin

GEMs represent cellular metabolic reactions to predict and understand metabolic behaviors. Their broad applicability includes cancer biology, metabolic engineering, biopharmaceutical production, and agriculture. Flux balance analysis (FBA) is the predominant method for solving GEMs, predicting reaction fluxes by maximizing or minimizing a metabolic objective. FBA imposes constraints using experimental data like metabolite exchanges to predict intracellular states. Despite advancements, GEMs, particularly for mammalian systems, face challenges due to being underdetermined and having limited constraints, making predictions unreliable.

The integration of exometabolomic data with GEMs can enhance model constraints, yet the complex relationships between external metabolites and internal cell states remain inadequately captured by traditional stoichiometric methods. Hybrid approaches combining mechanistic and data-driven models address these limitations by leveraging machine learning's predictive power alongside traditional models' biological insights. NEXT-FBA (Neural-net EXtracellular Trained Flux Balance Analysis) uses an ANN trained on CHO cell data to predict bounds for intracellular reactions. These bounds constrain a CHO cell GEM, accurately predicting metabolic states with minimal data. NEXT-FBA demonstrates

superior performance in predicting intracellular flux distributions over other traditional constraining methods.

This protocol outlines two steps in utilizing NEXT-FBA. The first step is if the user wishes to re-train the ANN on new datasets, either a completely new system (e.g. yeast, *E. coli*, HEK293) or adding data to the existing CHO cell database. This step is optional if the user does not wish to retrain the ANN. The second step deploys a pre-trained ANN to predict intracellular bounds and constrain a CHO cell GEM.

Download NEXT-FBA Toolkit

Timing: 5 minutes

1. Download NEXT-FBA files from https://github.com/J-Morrissey/NEXT-FBA (Figure 1). Click the green 'Code' button at the upper right corner and download the toolkit and the example dataset as a zip file by clicking 'Download Zip'.

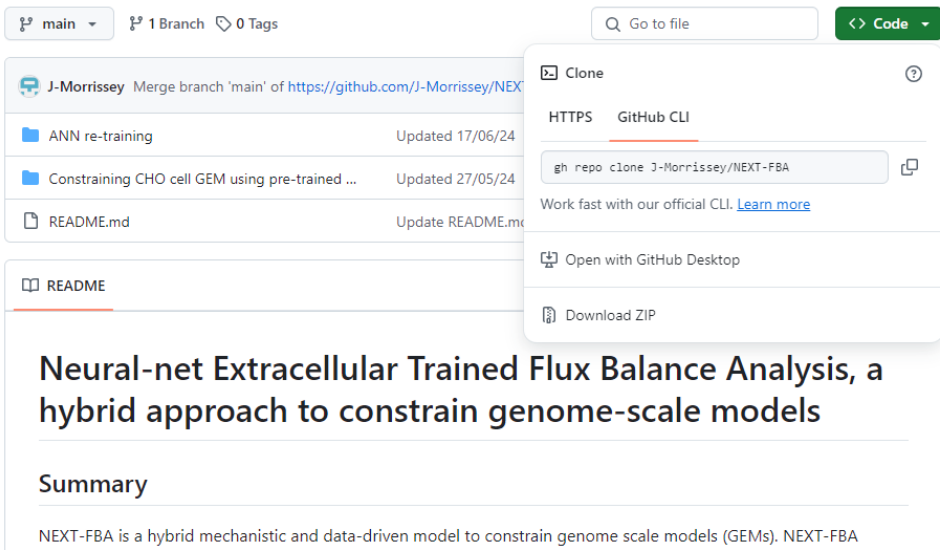


Figure 1: Screenshot of the NEXT-FBA GitHub page from which the files can be downloaded

Set up python working environment

Timing: 5 minutes

2. Open Python.
   Users must download and install Python beforehand. A Python integrated development environment (IDE), such as Pycharm, is recommended.
3. Unzip NEXT-FBA files into the working directory (Pycharm Project).

Formatting of user generated data

Timing: Variable (5 minutes if just using pre-trained ANN, 30 minutes for formatting data to train ANN with new data).

4. Formatting new dataset to train the ANN. Required training data are organized in four Excel files.
    a. Process-level data (exometabolomics): This file should include measurements of metabolite uptakes from the extracellular environment, growth rate and productivity. Organize the spreadsheet with observations labels along the first column and metabolite exchanges labels along the first row (example in Figure 2). The units for exchanges and productivity are mmol $gDCW^{-1}hr^{-1}$ and growth rate in hr^{-1}, where positive exchange means metabolite uptake and negative is secretion. This dataset can contain missing values as 'nan'.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Experiment	Ala	Antibody	Arg	Asn	Asp	Biomass	Cys	Glc	Gln	Glu	Gly	His	Ile	Lactate	Leu	Lys
2	EXPERIMENT_A_LB	-0,02557	0	0,004471	0,015558	-0,00876	0,00404167	-0,00036	0,238555	0,016273	0,003755	-0,01037	0,000715	0,001609	-0,25036	0,010372	0,002325
3	EXPERIMENT_A_UB	-0,01931	0	0,006617	0,01824	-0,00572	0,00683333	0,001609	0,267704	0,021102	0,014664	-0,00787	0,002504	0,006617	-0,18544	0,016631	0,004113
4	EXPERIMENT_A_MEDIAN	-0,02235	0	0,005544	0,01681	-0,00715	0,00541667	0,000715	0,25304	0,018777	0,009299	-0,00912	0,001609	0,004113	-0,21799	0,013412	0,003219
5	EXPERIMENT_B_LB	-0,02254	0	0,005338	0,018238	-0,00252	0,00625	-0,00015	0,155546	-0,00267	0,008155	-0,00297	0,00089	0,003114	-0,09401	0,014383	0,003559
6	EXPERIMENT_B_UB	-0,01839	0	0,007711	0,021797	0,000445	0,00825	0,002521	0,172005	-0,00089	0,0172	-0,00074	0,002669	0,008749	-0,0737	0,021352	0,005338
7	EXPERIMENT_B_MEDIAN	-0,02046	0	0,006524	0,020018	-0,00104	0,00725	0,001186	0,163849	-0,00178	0,012604	-0,00193	0,001779	0,005931	-0,08378	0,017794	0,004448
8	EXPERIMENT_C_LB	-0,05657	2,94753E-05	0,005498	0,009115	0,010561	0,00279167	0,001013	0,154225	-0,00405	0,047309	-0,00854	0,001302	0,007378	0,081742	0,018953	0,005498
9	EXPERIMENT_C_UB	-0,04659	3,54887E-05	0,00897	0,014902	0,0136	0,0045	0,004051	0,179977	-0,00145	0,058304	-0,00477	0,00434	0,014468	0,107928	0,02691	0,010127
10	EXPERIMENT_C_MEDIAN	-0,0515	3,24327E-05	0,007234	0,012008	0,012008	0,003625	0,002604	0,167101	-0,00275	0,052807	-0,00666	0,002894	0,010851	0,094907	0,023003	0,007813
11	EXPERIMENT_D_LB	-0,02169	1,83508E-05	0,004566	-0,001	0,004566	0,00341667	0,000143	0,142979	-0,00585	0,026969	-0,002	0,004281	0,003567	-0,05023	0,011416	0,003853
12	EXPERIMENT_D_UB	-0,01727	2,22761E-05	0,006564	0,001998	0,006421	0,00520833	0,001712	0,159104	-0,00086	0,034389	0	0,008134	0,008562	-0,02882	0,016267	0,006707
13	EXPERIMENT_D_MEDIAN	-0,01941	2,03134E-05	0,005565	0,000428	0,005422	0,00433333	0,000856	0,15097	-0,00328	0,030679	-0,001	0,006136	0,005993	-0,03953	0,013841	0,00528
14	EXPERIMENT_E_LB	-0,0853	4,48707E-05	0,00829	0,013743	0,015925	0,00279167	0,001527	0,232548	-0,00611	0,071335	-0,01287	0,001963	0,011126	0,123255	0,028578	0,00829
15	EXPERIMENT_E_UB	-0,07024	5,40249E-05	0,013525	0,022469	0,020506	0,0045	0,006108	0,271379	-0,00218	0,087914	-0,0072	0,006545	0,021815	0,16274	0,040576	0,015271
16	EXPERIMENT_E_MEDIAN	-0,07766	4,93728E-05	0,010908	0,018106	0,018106	0,003625	0,003927	0,251963	-0,00414	0,079625	-0,01003	0,004363	0,016361	0,143106	0,034686	0,01178

Figure 2: Demonstration of process-level data to input for the training of ANN.

    b. Intracellular flux data: This file should contain measurements of flux of intracellular reactions, recorded for each observation. Organize the spreadsheet with observation labels along the first column and intracellular reaction labels along the first row (example in Figure 3). The units of measurements for intracellular fluxes are mmol $gDCW^{-1}hr^{-1}$, where the positive values indicate flux directed according to reaction expression and negative opposite to reaction expression. This dataset can contain missing values as 'nan'.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Experiment	PGI	PFK	TPI	GAPDH	ENO	PK	HK	LDH	G6PDH	PPE	PPI	TKT1	TAL	TKT2
2	EXPERIMENT_A_LB	0,1402	0,211016	0,2103	0,456545	0,322067	0,322067	0,238555	-0,25036	0	-0,00107	0,000715	-0,00054	-0,00054	-0,00054
3	EXPERIMENT_A_UB	0,266094	0,2652	0,264664	0,529328	0,51681	0,51681	0,267704	-0,18544	0,111409	0,073498	0,037911	0,03666	0,03666	0,03666
4	EXPERIMENT_A_MEDIAN	0,203147	0,238197	0,237482	0,493026	0,419349	0,419349	0,25304	-0,21799	0,055615	0,036123	0,019313	0,018062	0,018062	0,018062
5	EXPERIMENT_B_LB	0,100682	0,138938	0,138049	0,293743	0,214561	0,214561	0,155546	-0,09401	0	-0,00133	0,001038	-0,00059	-0,00059	-0,00059
6	EXPERIMENT_B_UB	0,169929	0,168743	0,167853	0,336151	0,324733	0,324733	0,172005	-0,0737	0,060943	0,039442	0,021501	0,019721	0,019721	0,019721
7	EXPERIMENT_B_MEDIAN	0,135231	0,153915	0,153025	0,314947	0,269573	0,269573	0,163849	-0,08378	0,030397	0,019128	0,011269	0,00949	0,00949	0,00949
8	EXPERIMENT_C_LB	0,106481	0,143374	0,14294	0,299769	0,252749	0,252749	0,154225	0,081742	0	-0,00072	0,000434	-0,00029	-0,00029	-0,00029
9	EXPERIMENT_C_UB	0,178964	0,178385	0,177951	0,356047	0,320602	0,320602	0,179977	0,107928	0,060185	0,039641	0,020689	0,019821	0,019821	0,019821
10	EXPERIMENT_C_MEDIAN	0,14265	0,16088	0,160446	0,327836	0,286748	0,286748	0,167101	0,094907	0,030093	0,019387	0,010561	0,009693	0,009693	0,009693
11	EXPERIMENT_D_LB	0,05722	0,117295	0,116724	0,261844	0,196632	0,196632	0,142979	-0,05023	0	-0,00086	0,000571	-0,00043	-0,00043	-0,00043
12	EXPERIMENT_D_UB	0,157392	0,156821	0,15625	0,312643	0,293236	0,293236	0,159104	-0,02882	0,092751	0,061073	0,031535	0,030537	0,030537	0,030537
13	EXPERIMENT_D_MEDIAN	0,107306	0,136986	0,136558	0,287243	0,244863	0,244863	0,15097	-0,03953	0,046376	0,030108	0,016124	0,015126	0,015126	0,015126
14	EXPERIMENT_E_LB	0,160558	0,216187	0,215532	0,452007	0,381108	0,381108	0,232548	0,123255	0	-0,00109	0,000654	-0,00044	-0,00044	-0,00044
15	EXPERIMENT_E_UB	0,269852	0,268979	0,268325	0,536867	0,483421	0,483421	0,271379	0,16274	0,09075	0,059773	0,031195	0,029887	0,029887	0,029887
16	EXPERIMENT_E_MEDIAN	0,215096	0,242583	0,241928	0,494328	0,432373	0,432373	0,251963	0,143106	0,045375	0,029232	0,015925	0,014616	0,014616	0,014616

Figure 3 Demonstration of intracellular flux data to input for the training of ANN.

Note: Both exometabolomics and intracellular fluxes datasets should contain three data points for each experimental run, corresponding to the lower bound (LB), median (MEDIAN) and upper bound (UB) of the measured values.

c. Exometabolomics metadata: This file contains additional information related to process data, such as reaction equations, metabolite reference codes, etc. Organize the spreadsheet with labels of metadata information along the first column and extracellular metabolite exchange labels along the first row (example in Figure 4). This file has no mandatory fields. Any additional information regarding the extracellular exchange reactions can be included in this data file.

	A	B	C	D	E	F	G
1	Metabolite	Ala	Antibody	Arg	Asn	Asp	Biomass
2	Additional reaction information						

Figure 4: Demonstration of process-level metadata.

d. Intracellular flux metadata: This file contains additional information related to intracellular reactions, such as reaction expression and reversibility. Organize the spreadsheet with intracellular reaction labels along the first column, and the labels of metadata information along the first row (example in Figure 5). Ensure the presence of a column named 'Equilibrium', flagging reversible reactions (assign 1 for reversible reactions and 0 for irreversible ones). This file can include any additional information regarding intracellular reactions.

	A	B	C
1		Reaction	Equilibrium
2	PGI	G6P ↔ F6P	1
3	PFK	F6P → DHAP	0
4	TPI	DHAP ↔ GAI	1
5	GAPDH	GAP ↔ 3PG	1
6	ENO	3PG ↔ PEP	1
7	PK	PEP → Pyr	0
8	HK	Glc → G6P	0
9	LDH	Lac↔ Pyr	1
10	GALK	Gal → G6P	0
11	G6PDH	G6P → Ru5P	0
12	PPE	Ru5P ↔ X5P	1
13	PPI	Ru5P ↔ R5P	1
14	TKT1	X5P + R5P ↔	1

Figure 5: Demonstration of intracellular reaction metadata.

5. Formatting user generated process datasets for pre-trained ANNs.
   a. ***Input dataset into pre-trained ANN (data_to_predict)***: This dataset is used as an input into the ANN to make intracellular reaction bound predictions. A demonstration dataset 'Example NEXT-FBA Input Process Level Data.xlsx' is provided and should be replaced with a user dataset. Metabolite exchanges plus growth (biomass) and productivity (antibody) could be column labels with experiments in the rows, as shown in Figure 6. The units for exchanges and productivity are mmol $gDCW^{-1}hr^{-1}$ and growth rate in hr^{-1}, where positive exchange means metabolite uptake and negative is

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Experiment	Ala	Antibody	Arg	Asn	Asp	Biomass	Cys	Glc	Gln	Glu	Gly	His	Ile	Lactate	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
2	EXPERIMENT A	0.007382	0	0.001419	0.000563	0.009921	0.007665	-0.00024	0.118492	-0.00035	0.03004	-0.00354	0.000683	0.005639	0.055667	0.005152	0.001437	0.000959	0.001078	-0.04129	0.004623	0.002157	0.000427	0.000755	0.004216

Figure 6: Demonstration process level dataset to input into pre-trained ANN

secretion. The file can contain missing data and this will be automatically inputted into the ANN, see (Muñoz et al., 2004) for more details.

b. ***Input dataset to constrain GEM (uptake_data)***: This dataset contains the metabolite exchange data (uptake and secretions) that are used to constrain exchange reactions in the model. The dataset contains the same information as data_to_predict, it is just formatted to suit the GEM of choice. It is likely that the user has already formatted this dataset to fit their own constraining code, but a demonstration dataset for the iCHO2441 GEM, 'example_uptakes_iCHO2441.xlsx', is also provided. Row indexes are the iCHO2441 GEM name for the exchange reaction, and a column for the exchange lower bound, upper bound and median, as shown in Figure 7. The units for exchanges and productivity are mmol $gDCW^{-1}hr^{-1}$ and growth rate in hr^{-1}, where positive exchange means metabolite uptake and negative is secretion.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Experiment	experimentA_LB	experimentA_UB	experimentA_MEDIAN	experimentB_LB	experimentB_UB	experimentB_MEDIAN	experimentC_LB	experimentC_UB	experimentC_MEDIAN	experimentD_LB	experimentD_UB	experimentD_MEDIAN
2	EX_ala_L(e)	0.00017328	0.01678836	0.007382275	-0.019288247	-0.014918946	-0.017103597	-0.037739965	-0.030759162	-0.034249564	0.006586022	0.010349462	0.008467742
3	ICproduct_Final_d	0	0	0	3.94328E-05	4.70564E-05	4.32884E-05	2.26412E-05	2.79241E-05	2.53581E-05	3.07139E-05	3.66835E-05	3.36982E-05
4	EX_arg_L(e)	0.000760582	0.002956349	0.001419312	0.003964032	0.005800405	0.004882219	0.002399651	0.004799302	0.003490401	0.002284946	0.002822581	0.002553763
5	EX_asn_L(e)	0.000157407	0.000970899	0.000563492	0.030015198	0.037082067	0.033548632	0.036431065	0.047120419	0.041884817	0.001478495	0.006854839	0.004166667
6	EX_asp_L(e)	0.008994709	0.010846561	0.009920635	-0.012234043	-0.00933384	-0.010783941	0.008071553	0.012870855	0.010471204	0.006451613	0.011693548	0.009005376
7	biomass_cho	0.005925926	0.008724868	0.007665344	0.00705	0.009475	0.0082625	0.004083333	0.00775	0.005916667	0.00275	0.003458333	0.003083333
8	EX_cys_L(e)	-0.000445767	9.97354E-07	-0.000244709	0.005053191	0.006395643	0.005724417	0.003272251	0.006326353	0.004799302	-0.001209677	0.000537634	-0.000268817
9	EX_glc(e)	0.098593915	0.139440476	0.118492063	0.161600811	0.170276089	0.16593845	0.161212914	0.191317627	0.176265271	0.082123656	0.110483871	0.096370968
10	EX_gln_L(e)	-0.000669312	-3.1746E-05	-0.000350529	0.005091185	0.010663627	0.007877406	-0.014397906	-0.011998255	-0.013089005	-0.01061828	-0.008870968	-0.009677419
11	EX_glu_L(e)	0.026137566	0.033996032	0.030039683	-0.004305978	-0.001519757	-0.002912867	0.030104712	0.040575916	0.035340314	0.00577957	0.009005376	0.007392473
12	EX_gly(e)	-0.004466931	-0.002648148	-0.003537037	0.001139818	0.003900709	0.002520263	-0.008289703	-0.005017452	-0.006544503	-0.000403226	0.000134409	-0.000134409
13	EX_his_L(e)	0.000293651	0.001417989	0.00068254	0.001722391	0.002026342	0.001874367	0.004581152	0.007853403	0.006108202	0.001075269	0.002688172	0.00188172
14	EX_ile_L(e)	0.004365079	0.006912698	0.005638889	0.004356636	0.006281662	0.005319149	0.009598604	0.021160558	0.015488656	0.001612903	0.003494624	0.002553763
15	EX_lac_L(e)	0.049166667	0.063828042	0.055666667	0.024556738	0.039589666	0.032073202	-0.045375218	-0.018324607	-0.031849913	0.006048387	0.010080645	0.008064516
16	EX_leu_L(e)	0.00446164	0.005842593	0.005152116	0.009751773	0.014488349	0.012120061	0.004799302	0.010907504	0.007853403	0.005241935	0.00766129	0.006451613
17	EX_lys_L(e)	0.001239418	0.001634921	0.001436508	0.006648936	0.0079154	0.007282168	0.007417103	0.015052356	0.011343805	0.004166667	0.005913978	0.004973118
18	EX_met_L(e)	0.00077381	0.001148148	0.000956995	0.001621074	0.002596251	0.002108663	0.00109075	0.003272251	0.002181501	0.000806452	0.001344086	0.001075269
19	EX_phe_L(e)	0.000583333	0.001572751	0.001078042	0.003140831	0.004280648	0.00371074	0.003490401	0.006544503	0.005017452	0.002016129	0.002419355	0.002284946
20	EX_pro_L(e)	-0.137371693	0.079964286	-0.041293651	0.003622087	0.007700101	0.005661094	0.003490401	0.004799302	0.004144852	0.003225806	0.003629032	0.003494624
21	EX_ser_L(e)	0.003965608	0.005283069	0.004623016	0.009219858	0.011448835	0.010334347	0.010907504	0.016579407	0.013743455	0.007795699	0.010752688	0.009274194
22	EX_thr_L(e)	0.000914021	0.003486772	0.002157407	0.007155522	0.008092705	0.007624113	0.007635253	0.011125654	0.009380454	0.004032258	0.005376344	0.004704301
23	EX_trp_L(e)	0.000152116	0.000703704	0.000427249	0.001000507	0.002140324	0.001570415	0.00065445	0.002399651	0.001527051	0.000806452	0.001209677	0.001075269
24	EX_tyr_L(e)	0.000599206	0.000911376	0.000755291	0.002710233	0.003989362	0.003349797	0.006108202	0.009598604	0.007853403	0.002016129	0.002419355	0.002284946
25	EX_val_L(e)	0.003280423	0.005150794	0.004215608	0.007674772	0.008675279	0.008175025	0.012216405	0.019197208	0.015706806	0.005107527	0.00577957	0.005510753

Figure 7: Demonstration process level dataset to constrain your GEM. This contains exactly the same information as Figure 6 but allows the user to input lower and upper bounds, as well as bounds for additional metabolites not covered in Figure 6.

Key resources table

Key resources table

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
NEXT-FBA Master Folder	This article.	https://github.com/J-Morrissey/NEXT-FBA
ANN training data (optional). If the user wishes to re-train the ANN with new data, they must provide the extracellular exchange and the corresponding intracellular fluxes (from ^{13}C fluxomics or elsewhere)	User provided.	N/A

User provided process data to use in pre-trained ANN. This includes exchange data for metabolites (20 amino acids, lactate glucose), as well as growth rate and recombinant protein productivity, if appropriate.	User provided. Demonstration file available in GitHub.	https://github.com/J-Morrissey/NEXT-FBA
Software and algorithms		
Python 3.0	Python Software Foundation	https://www.python.org/downloads/
COBRApy	Python Package	Ebrahim et al., 2013
Pandas	Python Package	https://pandas.pydata.org/
Numpy	Python Package	https://numpy.org/
Tensorflow	Python Package	https://www.tensorflow.org
Scikit-learn	Python Package	https://scikit-learn.org/stable/
Scipy	Python Package	https://scipy.org/
Matplotlib	Python Package	https://matplotlib.org/
CPLEX	IBM	https://www.ibm.com/products/ilog-cplex-optimization-studio

Step-by-step method details

Here, the described step-by-step methods for re-training the NEXT-FBA ANN with user-provided datasets (Steps 1-3) and applying a trained ANN to a CHO cell GEM (Steps 4-5) are covered. Steps 1-3 are optional if the user does not wish to re-train the ANN and can immediately deploy the pre-trained ANN. Steps 4-5 can use either the pre-trained ANN from the master folder, or if the user is working with CHO cell systems, it can use the newly trained ANN from Steps 1-3.

Steps 4 and 5 are separated as it allows the user to take the intracellular bound predictions from Step 4 and apply them to another stoichiometric model other than iCHO2441. However, this would require a new back mapping file (similar to 'iCHO2441 Mapping.xlsx') and additional modifications not covered in this protocol. Likewise, if the user has trained the ANN with data from another cell system (e.g. *E. coli*, HEK293, yeast), this would require modifications to Steps 4-5 not covered in this protocol.

Training ANNs on new data

Timing: variable 1 hour and 35 minutes for each intracellular reaction

This step trains the ANNs on new user-provided datasets and sets up all requirements to predict bounds for intracellular fluxes used to constrain GEMs (Step Two).

1. Use the Python script "train_deploy_network_singlevariable.py" to train the ANNs on user-provided datasets. This step trains an ANN for a single intracellular reaction; hence, it should be repeated for each intracellular reaction available in the study.
    a. Specify the name of the main directory where the trained ANNs will be stored. This directory will contain the ANNs trained to predict the flux of each intracellular reaction.

```
>main_dir = r"Next-FBA"
```

    b. Choose the intracellular flux to be utilized for ANN training by setting the index of the desired intracellular reaction.

Note: In Python, indices start from 0 and range up to the total number of available intracellular reactions minus 1.

```
>variable_to_predict = 1
```

    c. Configure the inputs for data augmentation.
        i. Enable data augmentation by setting "do_smote" to "True"; set to "False" otherwise.
        ii. Specify the number of new observations to generate with SMOTE for each original observation using "sample_to_augment".
        iii. Specify the number of nearest neighbour observations to consider for SMOTE data augmentation (Chawla et al., 2002) with "neighbors_aug".
        iv. Specify the number of observations to generate by adding white noise (Arslan et al., 2019; Maharana et al., 2022) to each input observation using "noiseDAsamples".
        v. Set the amount of noise to add to the exometabolomics matrix with "x_noise" (specified as a percentage normalized to 1).
        vi. Set the amount of noise to add to the intracellular flux matrix with "y_noise" (specified as a percentage normalized to 1).

```
>do_smote = True
>sample_to_augment = 3
>neighbors_augm = 10
>noiseDAsamples = 5
>x_noise = 0.03
>y_noise = 0.01
```

    d. Specify the number of different ANN initializations to evaluate. The ANN exhibiting the lowest final loss value among these iterations will be chosen.

```
>n_models = 25
```

    e. Specify the file paths for input data. The paths of the four Excel files are provided as inputs to the "FluxSet" class, which manages the data.

Note: Please provide the absolute path for the data files.

```
>neuralFlux = FluxSet(
>       r"process_level_data.xlsx",
>       r"intracellular_flux_data.xlsx",
>       r"process_level_metadata.xlsx",
>       r"intracellular_flux_metadata.xlsx")
```

f. Specify for each intracellular reaction the activation function that will be used in the ANN layers. Populate the "`activation_list`" list with the chosen activation function (either "relu" or "tanh") for each intracellular reactions under analysis.

Note: Optimal activation function should be selected based on model performance in cross-validation.

g. Establish the search limits for optimizing the ANN hyperparameters (Negnevitsky, 2005). The "`nn1`" array specifies the number to explore for neurons of the first hidden layer. The "`nn2`" array specifies the number of neurons to explore for the second hidden layer. The "`lr`" array specifies the different learning rates to explore.

```
>nn1 = np.array([10, 20, 30, 40, 50])
>nn2 = np.array([0, 10, 20, 30, 40])
>lr = np.array([1e-2, 1e-3, 1e-4])
```

h. Configure parameters for ANN cross-validation. "`no_kfold_split`" specifies the number of splits for k-fold cross-validation used to identify the optimal hyperparameters. "`max_epochs`" specifies the maximum number of training epochs during cross-validation. "`cv_iterations`" specifies the number of iterations for the Monte Carlo cross-validation used to identify the optimal number of training epochs. "`mc_test_fraction`" specifies the fraction of observation allocated to the validation set within the Monte Carlo cross-validation process.

```
>no_kfold_split = 15
>max_epochs = 500
>cv_iterations = 100
>mc_test_fraction = 0.05
```

i. Execute the Python script to initiate ANN training.

j. The trained ANN for the chosen intracellular reaction is stored in a folder named "deployed_nn_#yourreactionname#" within the main NEXT-FBA directory (specified at step a). All additional necessary data is saved in the "utils" folder.

Note: Training time for the ANN varies depending on the hardware used. High-performance computers may be required for faster training.

2. Use the Python script "train_pca.py" to generate the PCA (Joliffe & Morgan, 1992) model necessary for assessing the similarity of new observations to the training dataset.

   a. Specify the name of the main NEXT-FBA directory where the trained ANNs are stored. The trained PCA model will be saved in this directory.

```
>main_dir = r"Next-FBA"
```

b. Specify the file path for input data. The paths of the four Excel files are provided as inputs to the "FluxSet" class, which manages the data.

Note: Please provide the absolute path for the data files.

```
>neuralFlux = FluxSet(
>       r"process_level_data.xlsx",
>       r"intracellular_flux_data.xlsx",
>       r"process_level_metadata.xlsx",
>       r"intracellular_flux_metadata.xlsx")
```

c. Configure parameters for PCA cross-validation, which are used to determine the optimal number of principal components (PCs). "max_pcs" defines the maximum number of PCs to test during cross-validation, while "cross_val_split" sets the number of k-fold data splits for cross-validation.

```
>max_pcs = 10
>cross_val_split = 8
```

d. Execute the Python script.

e. Review the results of the PCA cross-validation displayed in your Python terminal. Determine the optimal number of PCs either by selecting the configuration that minimizes the Root Mean Squared Error of Cross-Validation (RMSECV) or by employing the eigenvalue greater than one rule.

```
The selected intracellular flux is: PGI
Cross-validation results
   component       R2x     RMSEC    RMSECV
0          1  0.517405  0.662649  0.633048
1          2  0.682471  0.540292  0.540679
2          3  0.742972  0.488122  0.521502
3          4  0.796884  0.433735  0.494934
4          5  0.842945  0.381507  0.458785
5          6  0.876900  0.341038  0.421023
6          7  0.902309  0.304477  0.406096
7          8  0.923656  0.269848  0.370184
8          9  0.938928  0.241274  0.357472
9         10  0.951777  0.213378  0.341915
   component  eigenvalue       R2x   R2x tot
0          1   12.367721  0.515322  0.515322
1          2    3.907842  0.162827  0.678148
2          3    1.371650  0.057152  0.735300
3          4    1.325258  0.055219  0.790519
4          5    1.109272  0.046220  0.836739
5          6    0.832149  0.034673  0.871412
6          7    0.580314  0.024180  0.895592
7          8    0.542987  0.022624  0.918216
8          9    0.358838  0.014952  0.933168
9         10    0.307335  0.012806  0.945973
```

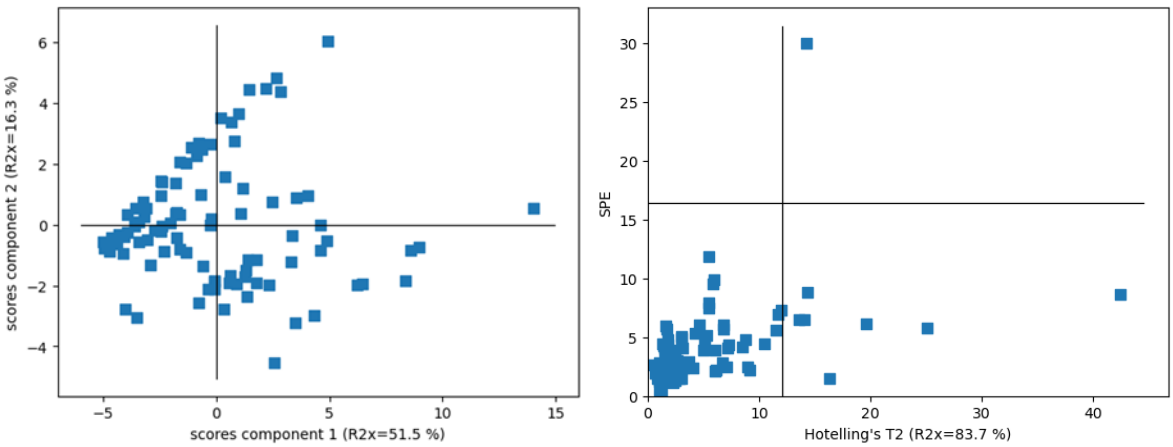
Figure 8: Example of the PCA cross-validation results displayed in your Python terminal. In this example, 5 PCs have been selected according to the eigenvalue greater than one rule.

f. Set the number of PCs determined in step 5 as "ncomp" in the script.

```
>ncomp = 5
```

g. Execute the Python script. Note: Computational time may increase with high number of observations or extensive process data.

h. The trained PCA model will be stored in the main NEXT-FBA directory. Additionally, plots depicting PCA scores (for the first and second PCs), an outlier map, and PCA loadings are generated. These images are not automatically saved, but users can do so if desired.



(a) (b)

(c)

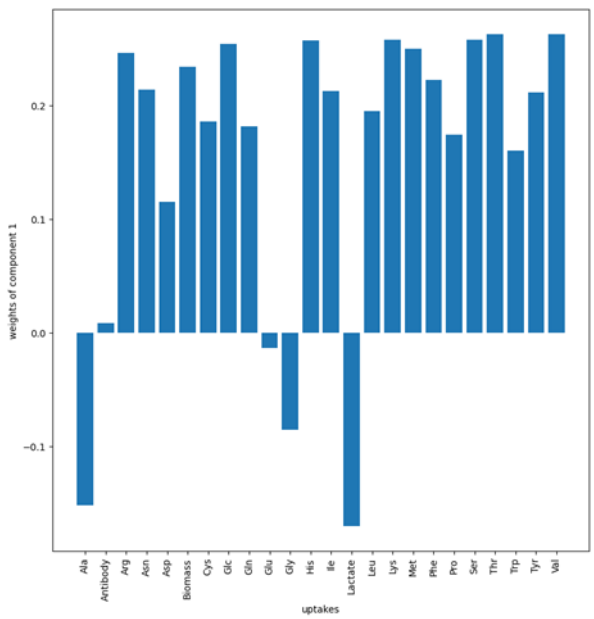


Figure 9: Example of plots depicting the PCA scores (a), the PCA outlier map (b), and the PCA loadings (c).

3. Use the Python script "create_flux_list.py" to generate the metadata which are essential for the ANNs to predict reaction bounds.
    a. Specify the name of the main NEXT-FBA directory where the trained ANN is stored. All generated metadata will be saved in this directory.

```
>main_dir = r"Next-FBA"
```

b. Specify the file path for input data. The paths of the four Excel files are provided as inputs to the "FluxSet" class, which manages the data.

Note: Please provide the absolute path for the data files.

```
>neuralFlux = FluxSet(
>       r"process_level_data.xlsx",
>       r"intracellular_flux_data.xlsx",
>       r"process_level_metadata.xlsx",
>       r"intracellular_flux_metadata.xlsx")
```

c. Execute the Python script.

d. Files containing exometabolomic input, available intracellular reactions, and reversible intracellular reactions are stored into the main NEXT-FBA directory.

Constraining CHO Cell GEM using pre-trained ANN

Timing: 20 minutes

This step uses a trained ANNs (either from previous step or from provided pre-trained ANN), to predict intracellular flux bounds using user-provided process data (Step 4). These intracellular bounds are then used to constrain the iCHO2441 GEM (Strain et al., 2023) (Step 5).

4. Use the python script 'Run Here ANN Intracellular Predictions.py' from the folder 'Step One Predicting Intracellular Bounds with Pre-trained ANN' to predict intracellular reaction bounds from the pre-trained ANN.
    a. Modify the inputs to the 'next_flux' function. Table 1 outlines each input.

```
>next_flux(data_to_predict, main_path, alpha=0.95,
>exclude_different=False, condition='both',
>condition_alpha ='95', >shsh=False, save_bounds=True,
>save_similarity=True)
```

Table 1: Inputs and output for pre-trained ANN.

Input	Description
data_to_predict	User provided process data, formatted as outlined above. This is a pandas.DataFrame with row and column index provided.
main_path	Path of the directory containing NEXT-FLUX files.
alpha	Confidence level of NN predicted intracellular fluxes. It is used to determine LB and UB.
condition	Rationale to define an experiment different from the training one. It is based on PCA projection. • 'T2': experiments outside Hotelling's T2 limit are excluded

	• 'SPE': experiments outside SPE limit are excluded • 'all': only experiments inside both limits are included • 'both' experiments outside both limits are excluded
condition_alpha	Select the confidence level of PCA diagnostics ('95'/'99')
shsh	Stop printing of info (True/False)
save_bounds	Flag for saving the predicted bounds as Excel file in the current directory (True/False)
`save_similarity`	Flag for saving the similarity scores as Excel file in the current directory (True/False)
Output	**Description**
predicted_bounds	Intracellular bounds predictions for 43 reactions in central carbon metabolism. This is used in the following steps to constrain a CHO cell GEM.
similarity	Similarity scores for new experiments

b. Run the script. This will generate two files.
   i. A 'predicted_bounds' excel file with the date and time. This file contains the predicted intracellular flux bounds used in Step 5.
   ii. A 'similarity' excel file with the date and time. This file will provide the similarity scores between the process data from the new experiment and the experiments used to train the ANN. It gives the distance from average and the difference in correlation. If these values are above 1 for any experiment, it differs significantly from the training dataset, and hence intracellular flux predictions may be unreliable.

5. Use the python script 'Run Here Constraining GEM.py' to constrain the CHO cell GEM with predicted intracellular bounds.
   a. Modify the inputs to the 'next_fba_constrained_model' function. Table 2 outlines each input.

```
>constrained_model = next_fba_constrained_model(model,
>experiment,mapping_table,uptake_data,
>trained_c13bound_data,solver)
```

Table 2: Inputs and outputs to next_fba_constrained_model function.

Input	**Description**
model	COBRApy metabolic model to constrain
experiment	Experiment from which to constrain model. This will select the correct bounds for exchange data and predicted bounds from ANN in Part One.
mapping_table	Back mapping table from NEXT-FBA reactions to iCHO2441 reactions. Provided in files.
trained_c13bound_data	Predicted bounds from ANN in Part One

Output	Description
constrained_model	NEXT-FBA constrained COBRApy metabolic model.

b. Run the script.
   i. This constrains 'model' to become 'constrained_model', a COBRApy model constrained with NEXT-FBA intracellular bounds.
   ii. This model is ready to be used for flux analysis, e.g. FBA or flux sampling. An example FBA growth maximization problem is given at the end of the script.

Expected outcomes

This protocol covers two major procedures for NEXT-FBA. The first procedure (Steps 1-3) is for users who wish to retrain the ANNs on new datasets. This will produce a set of files to allow the user to run the trained ANNs to predict intracellular fluxes for their study.

The second procedure (Steps 4-5) uses a pre-trained ANNs to predict intracellular flux bounds to constrain a CHO cell GEM. The pre-trained ANNs are provided in the files but could also use the newly trained ANNs from Steps 1-3. Step 4 will produce a set of intracellular flux bounds that are predicted from the user process data and Step 5 will use these bounds to constrain a CHO cell GEM. Step 4 will produce a constrained model, that can be used for flux analysis.

Limitations

The NEXT-FBA methodology has been designed to maximize its predictive capacity while working with a minimal amount of information. This simplification cannot fully capture or explain the biological mechanisms involved in generating the intracellular predictions. The process data input into the ANNs contains commonly measured metabolite data but does not consider any additional information about the process which would impact intracellular flux data. The reliability of this method depends entirely on the quality and quantity of data used to train the ANNs. This must be considered if the user wishes to re-train the ANNs with new data. NEXT-FBA predictions may suffer when the user process data differs significantly from the training dataset, the user is warned when the similarity scores are outside of the accepted range.

Troubleshooting

Problem 1:

While running the script 'train_deploy_network_singlevariable.py' an incorrect activation function might be given as input. The following message error appears:

```
> ValueError: Unknown activation function: #selected activation function#
```

Potential solution:

An unsupported activation function has been selected. Select one of the supported activation function: ‘relu’ and ‘tanh’.

Problem 2:

While running the script ‘train_deploy_network_singlevariable.py’ an empty array for hyperparameter grid search might be given as input. The following message error appears:

```
> ValueError: attempt to get argmin of an empty sequence
```

Potential solution:

An empty array of hyperparameter values to test has been provided. Provide an array of hyperparameter values with at least two elements.

Problem 3:

While running the script ‘train_deploy_network_singlevariable.py’ or ‘train_pca.py’ an excessive number of cross-validation data splits might be given as input. The following message error appears:

```
> ValueError: Cannot have number of splits n_splits=## greater than the
number of groups: ##
```

Potential solution:

The number of selected cross-validation data splits is greater than the number of observations groups; hence, the splitting of data cannot be accomplished. Reduce the number of data splits below the total number of groups acting on variables ‘no_kfold_split’ or ‘cross_val_split’.

Problem 4:

While running the script ‘train_pca.py’ or ‘train_pca.py’ an excessive number of principal components for cross-validation might be given as input. The following message error appears:

```
> ValueError: n_components=25 must be between 0 and min(n_samples,
n_features)=24 with svd_solver='full'
```

Potential solution:

The number of selected principal components for cross-validation is greater than rank of the data matrix. Reduce the number of principal components to test during cross-validation below the minimum between number of observations and number of extracellular exchange reactions, acting on the variable ‘max_pcs’.

Problem 5:

While running the script 'Run Here Constraining GEM.py'. There is an infeasibility during/after constraining the COBRApy model (Step 5). The following message error may appear:

```
>UserWarning: Solver status is 'infeasible'.
>warn(f"Solver status is '{status}'.", UserWarning)
```

Potential solution:

This error is due to infeasible constraints imposed on the model. Ensure that the process level dataset used to constrain the exchange reactions are feasible in the GEM before applying additional NEXT-FBA constraints. The most common fix is to test if the constrained growth rate is larger than the maximized growth rate. Test the constraints in an FBA problem to assess feasibility before applying the 'next_FBA_constraining_model' function.

Another source of this error is an issue with the MILP solver used in the 'next_FBA_constraining_model' function. The default GLPK lsolver for COBRApy fails to solve MILP correctly, which is why we recommend installing the CPLEX solver.

Problem 6:

While running the script 'Run Here Constraining GEM.py'. There is an error with the CPLEX solver (Step 5). The following message error may appear:

```
> cobra.exceptions.SolverNotFound
```

Potential solution:

Ensure that the CPLEX files are correctly installed and configured to Python in the system's environmental variables. For further information visit IBM's CPLEX download page (https://www.ibm.com/products/ilog-cplex-optimization-studio).

Resource availability

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Cleo Kontoravdi, (cleo.kontoravdi@imperial.ac.uk).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contacts, James Morrissey (rjm216@ic.ac.uk) and Gianmarco Barberi (gianmarco.barberi@unipd.it).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The processed example datasets and the pipeline codes are available at (https://github.com/J-Morrissey/NEXT-FBA). Zenodo DOI link: https://doi.org/10.5281/zenodo.13870919

Acknowledgments

J.M. thanks the UK Biological Sciences Research Council (BBSRC) and AstraZeneca for their funding and support. B.S would like to thank the UK BBSRC and GlaxoSmithKline for their funding and support. G.B. gratefully acknowledges the Foundation Ing. Aldo Gini (Padova, Italy) for his research scholarship during the period abroad at Imperial College London.

Author contributions

Conceptualization, data curation, methodology, writing, reviewing, editing J.M., G.M., B.S.
Reviewing, editing, supervision, funding acquisition P.F., C.K.

Declaration of interests

The authors have no competing interests.

References

Ebrahim, A., Lerman, J. A., Palsson, B. O., & Hyduke, D. R. (2013). COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Systems Biology*, *7*(74). https://doi.org/10.1186/1752-0509-7-74

Joliffe, I. T., & Morgan, B. (1992). Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research*, *1*(1). https://doi.org/10.1177/096228029200100105

Muñoz, S. G., Kourti, T., & MacGregor, J. F. (2004). Multivariate forecasting of batch evolution for monitoring and fault detection. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *37*(9). https://doi.org/10.1016/s1474-6670(17)31796-2

Strain, B., Morrissey, J., Antonakoudis, A., & Kontoravdi, C. (2023). How reliable are Chinese hamster ovary (CHO) cell genome-scale metabolic models? *Biotechnology and Bioengineering*, *120*(9). https://doi.org/10.1002/bit.28366

Figure legends

Figure 1: Screenshot of the NEXT-FBA GitHub page from which the files can be downloaded.

Figure 2: Demonstration of process-level data to input for the training of ANN.

Figure 3: Demonstration of intracellular flux data to input for the training of ANN.

Figure 4: Demonstration of process-level metadata.

Figure 5: Demonstration of intracellular reaction metadata.

Figure 6: Demonstration process level dataset to input into pre-trained ANN.

Figure 7: Demonstration process level dataset to constrain your GEM. This contains exactly the same information as Figure 6 but allows the user to input lower and upper bounds, as well as bounds for additional metabolites not covered in Figure 6.

Figure 8: Example of the PCA cross-validation results displayed in your Python terminal. In this example, 5 PCs have been selected according to the eigenvalue greater than one rule.

Figure 9: Example of plots depicting the PCA scores (a), the PCA outlier map (b), and the PCA loadings (c).

Table 1: Inputs and output for pre-trained ANN.

Table 2: Inputs and outputs to next_fba_constrained_model function.